



FAKULTÄT FÜR **INFORMATIK**

# Optimizing Shape Particle Filters for the Detection and Segmentation of Medical Images

DIPLOMARBEIT

zur Erlangung des akademischen Grades

**Diplom-Ingenieur**

im Rahmen des Studiums

**Medizinische Informatik**

eingereicht von

**Lukas Fischer BSc**

Matrikelnummer 0300795

an der

Fakultät für Informatik der Technischen Universität Wien

Betreuung:

a.o.Univ.-Prof. Dipl.-Ing. Dr.techn. Robert Sablatnig

Dipl.-Ing. Dr.techn. Georg Langs

Wien, 25.03. 2010

\_\_\_\_\_  
(Unterschrift Verfasser/in)

\_\_\_\_\_  
(Unterschrift Betreuer/in)

# Abstract

In recent years segmentation approaches based on sequential Monte Carlo Methods delivered promising results for the localization and delineation of anatomical structures in medical images. Also known as Shape Particle Filters, they are used for the segmentation of human vertebrae, lungs and hearts, being especially well suited to cope with the high levels of noise encountered in MR data and overlapping structures with ambiguous appearance in radiographs. Shape Particle Filters rely on a region template or map based on a shape models' mean shape, which is defined manually in existing approaches. During search, a classification step based on appearance features yields the probabilities for each pixel to belong to a certain region within the template. This forms the basis for the actual segmentation process.

This thesis aims at optimizing Shape Particle Filters in terms of computational performance as well as segmentation accuracy. Two novel approaches for the generation of the region map are proposed, namely automatic region maps and per-pixel region maps. The automatic region map approach, where the optimal distribution and number of template regions is derived from a set of training images, adapts to complex data and finds consistent features in the training examples without manual interaction. Using appearance features based on the Monogenic Signal the per-pixel region map approach eliminates both the need for the region estimation as well as the classification step, resulting in considerably faster segmentation while retaining the same level of accuracy.

The proposed methods are evaluated on four different data sets, synthetic rectangles, metacarpal bone radiographs, MRI slices of the heart and CT slices of the lung. Experimental results show a major gain in computational performance as well as better or at least equal segmentation results when compared to current approaches.

# Kurzfassung

In den letzten Jahren lieferten Segmentierungsansätze basierend auf sequenzielle Monte Carlo Methoden vielversprechende Ergebnisse bei der Lokalisierung und Beschreibung anatomischer Strukturen in medizinisch relevanten Bildern. Auch bekannt unter der Bezeichnung Shape Particle Filter wurden diese Methoden für die Segmentierung von Wirbelkörpern, Lungenflügeln und Herzen eingesetzt. Ihr großer Vorteil liegt darin, dass sie auch bei Bildern mit starkem Rauschen wie zum Beispiel MR Aufnahmen, sowie bei sich überlagernden Strukturen, bei denen eine eindeutige Unterscheidung der Objekte schwierig ist, noch sehr gute Segmentierungsergebnisse liefern. Shape Particle Filter benötigen eine Maske, welche auf dem Mean Shape eines Shape Models basiert und in existierenden Implementierungen immer manuell definiert wird. Während der Suche nach einem Objekt wird die Wahrscheinlichkeit für jeden Pixel zu einer gewissen Region der Maske zu gehören durch das Klassifizieren von Bildfeatures berechnet. Diese Wahrscheinlichkeiten bilden die Basis für den eigentlichen Segmentierungsprozess.

Das Ziel dieser Arbeit ist das Optimieren des bestehenden Shape Particle Filters um schnellere Laufzeiten und genauere Segmentierungsergebnisse erzielen zu können. Zwei neue Ansätze um die Masken zu generieren, automatische Masken und per-pixel Masken werden vorgestellt. Bei den automatischen Masken wird die optimale Anzahl und Anordnung der Maskenregionen aus den Features eines Trainingsbild Sets bestimmt. Die Methode passt sich komplexen Daten an und generiert Regionen aus Bereichen mit konsistenten, überlappungsfreien Bildfeatures. Eine Interaktion des Benutzers ist nicht notwendig. Die per-pixel Maske verwendet Bildfeatures basierend auf dem Monogenic Signal. Dadurch werden einerseits das Bestimmen der Regionen in der Maske und andererseits die Klassifizierung der Pixel unnötig. Dabei wird bei beträchtlich schnellerer Laufzeit die Segmentierungsgenauigkeit aufrecht erhalten.

Die vorgestellten Methoden wurden an Hand von vier unterschiedlichen Daten Sets evaluiert: synthetische Rechtecke, Handröntgenbilder, MRT Aufnahmen des Herzens und CT Aufnahmen der Lunge. Die Experimente zeigen eine signifikante Verbesserung

## *Kurzfassung*

der Laufzeit sowie bessere oder zumindest gleich gute Segmentierungsergebnisse im Vergleich zu bestehenden Implementierungen.

# Acknowledgements

I would like to thank everyone who helped me completing this thesis. First and most of all I want to thank Rene Donner for his expert advices, for hours and hours of time he spent helping me debugging code and unriddling my thoughts, for revising this thesis and for always being at hand when needed.

Furthermore I want to thank all my colleagues at the CIR Lab, especially Eva Dittrich for annotating the lungs data set, Ernst Schwarz for on the fly MATLAB and math theory support, Erich Birngruber for knowing everything, that is going on in my computer and how to tweak things the UNIX way, Werner Lang for modifying the ASM code to get results on my metacarpal bones data set and of course Georg Langs for co-supervising and revising this thesis. I also thank Professor Robert Sablatnig for the supervision of this thesis.

Additionally, I want to thank my parents for their imperturbable faith in my skills and for supporting me throughout my studies. I also want to thank Bernadette for being at my side for all this years and of course while writing this thesis. Thank you for your patience, sympathy and faith. Finally, I want to thank my daughter Aurelia for becoming the sunshine in my life!

# Contents

<b>Abstract</b>	<b>i</b>
<b>Kurzfassung</b>	<b>ii</b>
<b>Acknowledgements</b>	<b>iv</b>
<b>1. Introduction</b>	<b>1</b>
1.1. Motivation . . . . .	1
1.2. Aim of this Thesis . . . . .	3
1.3. Synopsis . . . . .	3
1.4. Nomenclature . . . . .	5
<b>2. Monte Carlo Methods</b>	<b>6</b>
2.1. Overview . . . . .	6
2.2. Markov Chain Monte Carlo (MCMC) . . . . .	9
2.2.1. Gibbs Sampling . . . . .	10
2.3. Sequential Monte Carlo Methods . . . . .	10
<b>3. Shape Models</b>	<b>12</b>
3.1. Principal Component Analysis . . . . .	12
3.2. Minimum Description Length . . . . .	14
3.3. Shape Models . . . . .	16
<b>4. Shape Particle Filter</b>	<b>19</b>
4.1. Preliminaries . . . . .	20
4.1.1. Gabor Jets . . . . .	21
4.1.2. $k$ -Nearest Neighbors . . . . .	23
4.2. Training . . . . .	24
4.3. Search . . . . .	26

<b>5. Optimizing Shape Particle Filters by Automated Training</b>	<b>28</b>
5.1. Preliminaries . . . . .	30
5.1.1. Monogenic Signal . . . . .	30
5.1.2. <i>kd</i> -Trees . . . . .	31
5.1.3. Support Vector Machines . . . . .	33
5.1.4. Differential Evolution . . . . .	36
5.2. Training . . . . .	38
5.3. Search . . . . .	43
<b>6. Experiments</b>	<b>46</b>
6.1. Setup . . . . .	48
6.1.1. Data Sets . . . . .	48
6.1.2. Testing Environment . . . . .	50
6.2. Results . . . . .	52
6.2.1. Residual Selection . . . . .	58
6.2.2. Automatic Region Determination . . . . .	59
6.2.3. Synthetic Rectangles . . . . .	62
6.2.4. Metacarpal Bones . . . . .	64
6.2.5. Hearts . . . . .	67
6.2.6. Lungs Left . . . . .	70
6.2.7. Lungs Right . . . . .	73
6.3. Comparison with ASMs and AAMs . . . . .	76
<b>7. Conclusion and Outlook</b>	<b>78</b>
7.1. Conclusion . . . . .	78
7.2. Outlook . . . . .	80
<b>A. Additional Material and Results</b>	<b>81</b>
A.1. Shape Models: Eigenmodes and Eigenvalues of the Used Data Sets . . . . .	81
<b>Bibliography</b>	<b>86</b>

# Chapter 1.

## Introduction

Medical image segmentation aims at segmenting known anatomic structures in medical images. Such structures are e.g. organs, such as lungs, cardiac ventricles, as well as bones, vessels or brain structure. Methods applied in this field are generally application-specific, that means they use prior knowledge of the object of interest as well as other surrounding structures (e.g. prominent anatomical structures like bones or organs) in the image. The segmentation of objects of interest in images on the one hand offers assistance in the localization and on the other hand allows to draw conclusions about abnormal or pathological conditions of the respective objects. Shape Particle Filters for medical image segmentation provide the possibility to locate and segment objects without user interaction. However the challenge in this field lies within the improvement of the localization and segmentation accuracy.

In Sec. 1.1 the motivation, as well as the medical background are outlined. Sec. 1.2 describes the aim of the thesis. A brief outline of this thesis is given in Sec. 1.3 and the nomenclature used is depicted in Sec. 1.4.

### 1.1. Motivation

Statistical methods such as sequential Monte Carlo Methods were proposed for segmentation [Lee03, Florin05, deBruijne06, Seise09] and tracking [Doucet01, Arulampalam02, Giebel04, Liebelt07, Sørensen08, Wörz09] of objects.

A similar approach, called Shape Particle Filters was introduced in [deBruijne04a, deBruijne04b, deBruijne05] for the segmentation of vertebræ, lungs and hearts. In



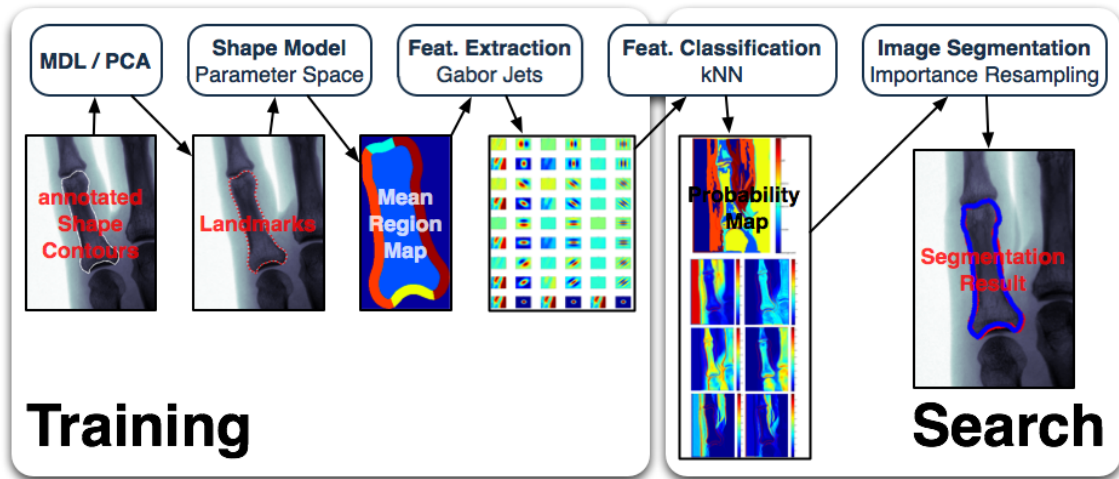


Figure 1.1.: Flowchart of a Shape Particle Filter. A detailed description of all parts of the filtering pipeline is provided in Chapter 4.

Fig. 1.1 an outline of a Shape Particle Filter approach is depicted: Based on a global shape model a *labeling template* or *region map* for the following feature extraction is defined according to the respective medical object of interest (e.g. interior of the contour and one or more regions within a certain border on the outside).

The number and location of these regions can be defined in two ways e.g. manually [deBruijne04a, deBruijne04b] or automatically (see Sec. 5.2). Using these regions and their corresponding distributions in the feature space Particle Filtering approaches are used to estimate the most probable point in the shape parameter space (corresponding to a segmentation of the image) for a given test image. This is achieved by sampling from the image features according to shape hypothesis and computing corresponding confidence values. Estimating these posterior probabilities over the parameter space allows to find values of maximum confidence, i.e. to optimize the fit of the model to the object in the test image.

Existing Shape Particle Filtering approaches suffer from two major drawbacks:

1. **Computational Performance/Speed**

Depends mainly on the methods used for the image feature extraction (see Sec. 4.2 and 5.2) and image feature classification steps (see Sec. 4.3 and 5.3).

2. **Segmentation Accuracy**

Depends on the suitability of the extracted features and on the region map based on these features.

## 1.2. Aim of this Thesis

The goal of this thesis was to improve existing 2D Shape Particle Filter approaches for medical image segmentation with regard to computational performance and accuracy. This is achieved by the following three contributions

1. Introduction of an automatic region estimation, yielding more accurate results, as well as avoiding the bias incorporated by manual region map definition.
2. Elimination of the computationally expensive classification and the region map estimation by directly computing per-pixel similarity measures. This is made possible by using appearance features based on the Monogenic Signal (as presented in Sec. 5.1.1) leading to faster and more accurate segmentation results.
3. A comparison to different segmentation approaches like the well known Active Shape Models (ASMs) [Cootes92] and Active Appearance Models (AAMs) [Cootes01].

## 1.3. Synopsis

This thesis is subdivided into four major parts: a survey of sequential Monte Carlo methods, a detailed theoretical background on Shape Models, a description of the original Shape Particle Filter and a description of the improved Shape Particle Filtering scheme developed within the scope of this thesis.

A survey of sequential Monte Carlo methods is given in Chapter 2 including the historical background and the state of the art in the field of Monte Carlo methods. Furthermore Markov Chain Monte Carlo including Gibbs sampling and Particle Filters, also known as sequential Monte Carlo methods, are outlined.

Throughout Chapter 3 to Chapter 5 two medical data sets are used to illustrate the results. These are metacarpal bone radiographs and MRI slices of the heart (Fig. 1.2). A detailed description of the two data sets can be found in Sec. 6.1.1.

In Chapter 3 the fundamental methods needed in shape modeling e.g. Principle Component Analysis (PCA) and Minimum Description Length (MDL) as well as the derivation of Shape Models are described.

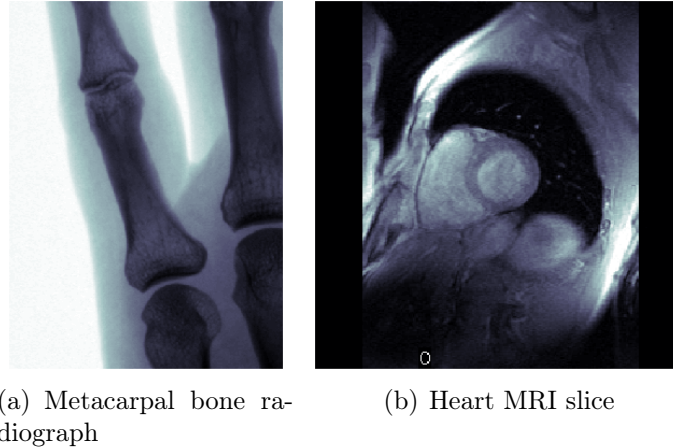


Figure 1.2.: Example images of the two data sets used to illustrate preliminary results throughout the theoretical part of this thesis. A detailed description can be found in Sec. 6.1.1.

The Shape Particle Filter introduced in [deBruijne04a, deBruijne04b] in the medical context is detailed in Chapter 4. All important parts of the approach, namely the employed region map, the feature extraction and classification schemes as well as the actual segmentation are presented.

Chapter 5 contains the main contribution of this thesis. It is similarly structured as the previous chapter presenting the improved region map computation approaches, the refined feature extraction and the optimized classification schemes as well as the actual segmentation.

The evaluation of the proposed methods and details of the data sets and the experimental setup are reported in Chapter 6. Four 2D data sets, synthetic rectangles, metacarpal bones, hearts and left and right lungs were used for evaluation. For each data set five segmentation approaches based on Shape Particle Filters are evaluated by Leave One Out Cross Validation and presenting the segmentation results, as well as the respective landmark errors and a comparison of the segmentation runtime.

Finally a conclusion and an outlook on future work are given in Chapter 7.

Parts of the method proposed in the scope of this thesis were published and presented at the Workshop on Probabilistic Models for Medical Image Analysis (PMMIA 2009) at the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI 1009), London (UK) [Fischer09]. Parts were also presented at the European Congress of Radiology (ECR) 2010, Vienna, Austria.

## 1.4. Nomenclature

Symbol	Description
<b>General</b>	
iid	independent and identically-distributed random variables
$\mathbf{x}$	vector representation
$\mathbf{X}$	matrix representation
$\bar{\mathbf{x}}$	mean vector $x$
$p(\mathbf{x})$	probability distribution of samples $\mathbf{x}$
$\mathbf{e}_i$	eigenvector
$\lambda_i$	eigenvalue
$\sigma^2$	variance
$d$	number of dimensions
$n$	number of landmarks per shape
$N_S$	number of shapes
$N_P$	number of pixels in an image
$N_J$	number of pixels per region
$\mathbf{v}_i$	point position vector of length $n$ times $d$
<b>Monte Carlo</b>	
$N$	number of samples
$\mathbf{x}_i$	set of $i = 1, \dots, N$ samples
$\mathbf{s}_i$	markov chain state at step $i = 1, \dots, N$
<b>PCA</b>	
$m$	number of data samples
$\mathbf{A}$	$m \times nd$ sample matrix
$\mathbf{C}$	covariance matrix
$\mathbf{E}$	feature vector/matrix containing the first $p$ eigenvectors
<b>MDL</b>	
$\mathcal{H}$	set of Hypothesis
$H$	hypothesis
$L(H)$	hypothesis' $H$ description length
$L(D H)$	length of the description of data $D$ encoded with the hypothesis $H$
$\text{COMP}(\mathcal{H})$	parametric complexity
$\mathbf{c}_i$	contour vector $i = 1, \dots, N_S$
<b>Shape Model</b>	
$\mathbf{b}$	model parameter vector
$\mathbf{t}$	similarity transformation parameter vector
$\mathbf{c}$	combined parameter vector
<b>Particle Filter</b>	
$l$	label of a region in region map
$N_L$	number of region map regions $N_L = N_{inner} + N_{border}$
$\pi(\mathbf{c}_i)$	likelihood of shape represented by combined parameter vector $\mathbf{c}_i$
$\mathbf{f}$	average feature vector
$\mathbf{F}$	feature matrix
$N_{inside}$	number of pixels inside the shape

# Chapter 2.

## Monte Carlo Methods

This chapter gives a brief overview of Monte Carlo methods and their sequential variants, namely Markov Chain Monte Carlo and Particle Filters. Sec. 2.1 concentrates on the Monte Carlo method itself and also includes the description of two simple sampling algorithms, rejection sampling and importance sampling. In Sec. 2.2 the basics of Markov Chain Monte Carlo including Markov Chain theory and Gibbs sampling are described. Particle Filters also known as sequential Monte Carlo methods are depicted in Sec. 2.3.

### 2.1. Overview

Monte Carlo techniques can be traced back to Babylonian and Old Testament times<sup>1</sup> [Hammersley64]. Later descriptions of Monte Carlo experiments include for example Buffon's needle experiment and also played a major role in the simulation of the Manhattan project [Andrieu03].

[Halton70] defined the Monte Carlo method as *representing the solution of a problem as a parameter of a hypothetical population and using a random sequence of numbers to construct a sample of the population, from which statistical estimates of the parameter can be obtained.*

---

<sup>1</sup>An early mathematical experiment on the numerical value of  $\pi$  can be found in the Old Testament (1 Kings vii. 23 and 2 Chronicles iv. 2). People observed that the columns of King Solomon's temple were about three times as great in circumference as thickness. If they inferred this as an universal property of circular objects is not conveyed. [Hammersley64]

These techniques are usually applied in mathematical scenarios, where no closed-form solution can be calculated for a certain probability distribution  $p$  due to the problems complexity. Considering the definition above the principle of Monte Carlo Methods can be described by replacing the algebraic representation of  $p$  (e.g. a Gaussian:  $\frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}x^2)$ ) with a *random sample* or *population* representation of  $p$  (e.g. a set of samples  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N \stackrel{iid}{\sim} p(\mathbf{x}) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}x^2)$ ) [Andrieu03]. In other words the problem is to find the expectation  $\mathbb{E}$  of a function  $f(\mathbf{x})$  with respect to a probability distribution  $p(\mathbf{x})$

$$\mathbb{E}(f(\mathbf{x})) = \int f(\mathbf{x})p(\mathbf{x})d\mathbf{x} \quad (2.1)$$

Obvious Monte Carlo Methods are based on numerical sampling, where the general idea is to independently draw a set of samples  $\mathbf{x}_i$  ( $i = 1, \dots, N$ ) from the distribution  $p(\mathbf{x})$ . In this way the expectation in Eq. 2.1 can be approximated by a finite sum, forming the estimator

$$\hat{f} = \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}_i) \quad (2.2)$$

$\mathbb{E}(\hat{f}) \approx \mathbb{E}(f)$  and therefore  $\hat{f}$  has the correct mean as long as all samples  $\mathbf{x}_i$  are drawn from  $p(\mathbf{x})$ . The estimator's variance is then

$$\begin{aligned} \sigma^2(\hat{f}) &= \sigma^2 \left[ \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}_i) \right] \\ &= \frac{1}{N} \sigma^2[f(\mathbf{x})] \\ &= \frac{1}{N} \mathbb{E}[(f - \mathbb{E}(f))^2] \end{aligned} \quad (2.3)$$

which equals the variance of  $f(\mathbf{x})$  under  $p(\mathbf{x})$ . Therefore the accuracy of the estimator does not depend on the dimensionality of  $\mathbf{x}$  [Bishop07].

For an extensive description and derivation of Monte Carlo Methods please refer to [Halton70, Gordon93, Doucet01, MacKay03, Andrieu03, Bishop07].

In the following the two most common sampling methods in the scope of Monte Carlo Methods, namely *rejection sampling* and *importance sampling* are briefly described.

**Rejection Sampling** For the description of this sampling method a simple distribution  $q(\mathbf{x})$  enveloping a more complex  $p(\mathbf{x})$  than the Gaussian used in the previous equations is introduced. Furthermore a constant  $k$  is used to ensure that  $kq(\mathbf{x}) \geq p(\mathbf{x})$ , where  $kq(\mathbf{x})$  is called comparison function. Then two random numbers are sampled,  $x$  from  $q(\mathbf{x})$  and  $u$  from a uniform distribution  $U[0, kq(\mathbf{x})]$ . If  $u < \frac{p(x)}{kq(x)}$ ,  $x$  is accepted, otherwise it is rejected and the sampling is repeated. Formally the accepted samples are given by

$$\begin{aligned} p(\text{accept}) &= \int \frac{p(x)}{kq(x)} q(x) dx \\ &= \frac{1}{k} \int p(x) dx. \end{aligned} \tag{2.4}$$

That means samples are rejected if they fall between the distribution  $p(x)$  and  $kq(x)$ . The success of this approach strongly depends on how well the simpler distribution  $q(\mathbf{x})$  fits the intended distribution  $p(\mathbf{x})$ .

Extensions based on rejection sampling include *adaptive rejection sampling* as well as the *Metropolis-Hastings algorithm* [Metropolis53, Bishop07].

**Importance Sampling** This technique allows to evaluate a function's expectation value with respect to a certain complex distribution  $p(\mathbf{x})$ . Similar to rejection sampling samples are drawn from a simpler distribution  $q(\mathbf{x})$ . Considering a function  $f(\mathbf{x})$ , its expectation can therefore be denoted as a finite sum of samples  $\mathbf{x}_i$  ( $i = 1, \dots, N$ ) drawn from  $q(\mathbf{x})$

$$\begin{aligned} \mathbb{E}(f(x)) &= \int f(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} \\ &= \int f(\mathbf{x}) \frac{p(\mathbf{x})}{q(\mathbf{x})} q(\mathbf{x}) d\mathbf{x} \\ &\cong \frac{1}{N} \sum_{i=1}^N \frac{p(\mathbf{x}_i)}{q(\mathbf{x}_i)} f(\mathbf{x}_i) \end{aligned} \tag{2.5}$$

where  $\frac{p(\mathbf{x}_i)}{q(\mathbf{x}_i)}$  are called *importance weights*. In difference to rejection sampling all samples are retained [Bishop07].

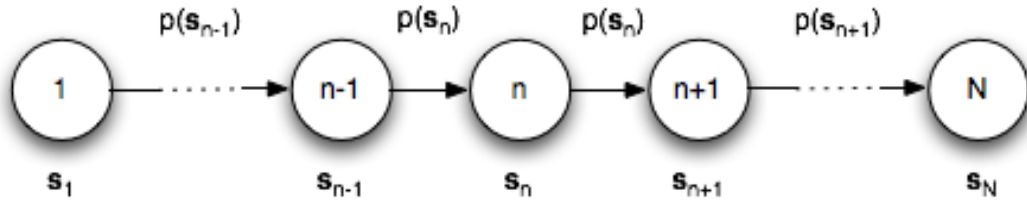


Figure 2.1.: Markov chain example. Nodes represent the  $\mathbf{s}_i$  states and edges represent the transition probabilities  $p_{ij}$  to move from  $\mathbf{s}_i$  to  $\mathbf{s}_j$  ( $i, j = 1, \dots, N$ ).

## 2.2. Markov Chain Monte Carlo (MCMC)

The MCMC framework encompasses sampling algorithms, that allow sampling of a large class of distributions and are based on the construction of a Markov chain to approximate a target distribution. Such a chain can be formulated as a sequence of random variables  $\mathbf{s}_1, \dots, \mathbf{s}_N$  where the future and past states are independent under a given present state such that the following conditional independence property holds for  $n \in \{1, \dots, N-1\}$

$$p(\mathbf{s}_{n+1} | \mathbf{s}_1, \dots, \mathbf{s}_n) = p(\mathbf{s}_{n+1} | \mathbf{s}_n). \quad (2.6)$$

The state space of the chain is a countable set  $\mathcal{S}$  formed by the possible values for  $\mathbf{s}_i$ . Directed graphs like the one in Fig. 2.1 can be used for the description of Markov chains [Bishop07].

By observing the chain's state after an adequate number of steps, samples can be drawn to represent the target distribution. With increasing number of steps the quality of the samples improves. Different algorithmic approaches exist and can be roughly divided in two groups: random walk algorithms and more sophisticated algorithms avoiding random walk.



### 2.2.1. Gibbs Sampling

Gibbs sampling [Geman84] is a Markov Chain Monte Carlo algorithm and furthermore a special case of the Metropolis-Hastings Algorithm [Metropolis53]. It is also known as *heat bath method* or ‘Glauber dynamics’ [MacKay03].

Consider a distribution  $p(\mathbf{x}) = p(x_1, \dots, x_N)$  that is too complex to sample from directly and a Markov chain similar to Eq. 2.6. Furthermore consider the conditional distributions  $p(x_i | \{x_j\}_{j \neq i})$  ( $i, j = 1, \dots, N$ ) where  $x_j$  denotes  $x_1, \dots, x_N$  without  $x_i$  and it is easier to sample from. Starting from the current state  $\mathbf{s}^{(t)}$  of the algorithm at step  $t$ ,  $x_i^{(t)}$  is sampled from the conditional density  $p(x_i | x_1^{(t)}, \dots, x_N^{(t)} \setminus x_i^{(t)})$  leading to a new value  $x_i^{(t+1)}$ . This procedure is then repeated by selecting the variables iteratively or randomly and sampling one parameter at a time leading to a new state  $\mathbf{s}^{(t+1)}$  [MacKay03, Bishop07]:

$$\begin{aligned}
 x_1^{(t+1)} &\sim p(x_1 | x_2^{(t)}, x_3^{(t)}, \dots, x_N^{(t)}) \\
 x_2^{(t+1)} &\sim p(x_2 | x_1^{(t+1)}, x_3^{(t)}, \dots, x_N^{(t)}) \\
 &\vdots \\
 x_n^{(t+1)} &\sim p(x_n | x_1^{(t+1)}, \dots, x_{n-1}^{(t+1)}, x_{n+1}^{(t)}, \dots, x_N^{(t)}) \\
 &\vdots \\
 x_N^{(t+1)} &\sim p(x_N | x_1^{(t+1)}, x_2^{(t+1)}, \dots, x_{N-1}^{(t+1)})
 \end{aligned} \tag{2.7}$$

### 2.3. Sequential Monte Carlo Methods

Particle filtering was introduced with the intention to implement recursive Bayesian filters [Gordon93]. It is also known as Sampling Importance Resampling (SIR), Bayesian bootstrap filter or *sequential Monte Carlo Methods*. In contrast to other filters that use Monte Carlo Methods to get estimates of the mean and covariance of the posterior, Particle Filters approximate the complete posterior. They aim at approximating posterior densities using swarms of points (so called *particles*) in a sample space. A weight is assigned to each particle and using a discrete distribution of the particles the posterior distribution can be approximated. This results in particle probabilities which are proportional to the particle weights. The algorithms differ mainly in the way how the particle swarms evolve and adapt to input data [Fearnhead08].

Formally Particle Filters can be described as follows: Considering a class of distributions represented by a Markov chain, where for each state  $\mathbf{s}_i$ ,  $i = 1, \dots, N$ , an observation  $\mathbf{X}_i = (\mathbf{x}_1, \dots, \mathbf{x}_i)$  is given. The goal is to draw  $M$  samples from the posterior distribution  $p(\mathbf{s}_i|\mathbf{X}_i)$ . Applying Bayes' Theorem leads to the expectation of the state function  $f(\mathbf{s}_i)$

$$\begin{aligned}
 \mathbb{E}(f(\mathbf{s}_i)) &= \int f(\mathbf{s}_i)p(\mathbf{s}_i|\mathbf{X}_i)d\mathbf{s}_i \\
 &= \int f(\mathbf{s}_i)p(\mathbf{s}_i|\mathbf{x}_i, \mathbf{X}_{i-1})d\mathbf{s}_i \\
 &= \frac{\int f(\mathbf{s}_i)p(\mathbf{x}_i|\mathbf{s}_i)p(\mathbf{s}_i|\mathbf{X}_{i-1})d\mathbf{s}_i}{\int p(\mathbf{x}_i|\mathbf{s}_i)p(\mathbf{s}_i|\mathbf{X}_{i-1})d\mathbf{s}_i} \\
 &\cong \sum_{m=1}^M w_i^{(m)} f(\mathbf{s}_i^{(m)})
 \end{aligned} \tag{2.8}$$

where the samples  $\mathbf{s}_i^{(m)}$  are drawn from  $p(\mathbf{s}_i|\mathbf{X}_{i-1})$  [Bishop07]. The weights  $w_i^{(m)}$  are defined by

$$w_i^{(m)} = \frac{p(\mathbf{x}_i|\mathbf{s}_i^{(m)})}{\sum_{l=1}^M p(\mathbf{x}_i|\mathbf{s}_i^{(l)})} \tag{2.9}$$

and are normalized ( $0 \leq w_i^{(m)} \leq 1$  and  $\sum w_i^{(m)} = 1$ ).

Starting at a time step  $i$ , where the weights and samples are already obtained, the next value  $\mathbf{x}_{i+1}$  is observed. Samples and weights at time step  $i+1$  can then be drawn from the distribution  $p(\mathbf{s}_{i+1}|\mathbf{X}_i)$

$$\begin{aligned}
 p(\mathbf{s}_{i+1}|\mathbf{X}_i) &= \int p(\mathbf{s}_{i+1}|\mathbf{s}_i, \mathbf{X}_i)p(\mathbf{s}_i|\mathbf{X}_i)d\mathbf{s}_i \\
 &= \int p(\mathbf{s}_{i+1}|\mathbf{s}_i)p(\mathbf{s}_i|\mathbf{X}_i)d\mathbf{s}_i \\
 &= \int p(\mathbf{s}_{i+1}|\mathbf{s}_i)p(\mathbf{s}_i|\mathbf{x}_i, \mathbf{X}_{i-1})d\mathbf{s}_i \\
 &= \frac{\int p(\mathbf{s}_{i+1}|\mathbf{s}_i)p(\mathbf{x}_i|\mathbf{s}_i)p(\mathbf{s}_i|\mathbf{X}_{i-1})d\mathbf{s}_i}{\int p(\mathbf{x}_i|\mathbf{s}_i)p(\mathbf{s}_i|\mathbf{X}_{i-1})d\mathbf{s}_i} \\
 &\cong \sum w_i^{(m)} p(\mathbf{s}_{i+1}|\mathbf{s}_i^m)
 \end{aligned} \tag{2.10}$$

which is a mixture distribution. Consequently the representation for the next time step is obtained by drawing  $M$  samples from the mixture distribution (Eq. 2.10) and evaluating the weights  $w_{n+1}^{(m)}$  by using the observation  $\mathbf{x}_{i+1}$ . A detailed derivation of the above described equations can be found in [Bishop07].

# Chapter 3.

## Shape Models

This chapter is about the fundamental methods needed to create a shape model of a given set of manually annotated shapes (i. e., their contour). Sec. 3.1 describes PCA, that allows to reduce the dimensionality of the data and provides its eigenvectors, which are later needed for the Shape Model generation. In order to obtain equally distributed landmarks of a set of shapes, Minimum Description Length as described in Sec. 3.2 is applied. In Sec. 3.3 point distribution Shape Models are described.

### 3.1. Principal Component Analysis

PCA is a well known and widely applied technique in the field of Pattern Recognition, Image Understanding and Computer Vision. PCA is mainly used for dimensionality reduction, feature extraction, lossy data compression and data visualization. It is also known as the *Karhunen-Loeve* transform, the *Hotelling* transform or proper orthogonal decomposition [Bishop07]. PCA was first introduced by Karl Pearson in 1901 [Pearson01]. He defined PCA as the linear projection that minimizes the average projection cost, in other words the mean squared distance between the data points and their projection [Pearson01]. Besides this definition from Pearson there exists a second also commonly used definition formulated by Hotelling, defining the PCA as the orthogonal projection of data onto a lower dimensional linear space, which maximizes the variance of the projected data.

Consider an  $nd$ -dimensional space (e.g. 2D or 3D) and the containing data represented as a  $m \times nd$  Matrix  $\mathbf{A}$ , whereas  $m$  donates the number of samples, i. e. the number of

shapes, and  $nd$  denotes the number of landmarks times their dimensionality. In other words each column represents data dimensions containing a vector of length  $m$ , short  $\mathbf{A}_i = (\mathbf{A}_{1i}, \dots, \mathbf{A}_{mi})^T$ ,  $i \in \{1, \dots, nd\}$ . To produce a normalized data set (with zero mean), the mean  $\bar{\mathbf{A}}_i$  of each dimension or column  $\mathbf{A}_i$  is subtracted, short  $\mathbf{A}_i^* = \mathbf{A}_i - \bar{\mathbf{A}}_i$  with  $j \in \{1, \dots, m\}$ . Normalization is crucial for PCA to assure that the data is centered around the origin, otherwise a wrong direction for the eigenvectors is calculated. The next step is to calculate the covariance matrix of the data, which for  $nd$ -dimensional data is defined as follows [Smith02]:

$$\mathbf{C}^{nd \times nd} = c_{i,j}, \quad c_{i,j} = cov(\mathbf{A}_i, \mathbf{A}_j) \quad (3.1)$$

In matrix notation ( $i \in \{1, \dots, nd\}, j \in \{1, \dots, m\}$ ):

$$\mathbf{C} = \begin{pmatrix} cov(\mathbf{A}_1, \mathbf{A}_1) & \cdots & cov(\mathbf{A}_1, \mathbf{A}_j) \\ \vdots & \ddots & \vdots \\ cov(\mathbf{A}_i, \mathbf{A}_1) & \cdots & cov(\mathbf{A}_i, \mathbf{A}_j) \end{pmatrix} \quad (3.2)$$

Now the eigenvectors  $\mathbf{e}_i$  and their eigenvalues  $\lambda$  are calculated, resulting in  $\min(nd, m)$  eigenvectors. The eigenvector with the largest eigenvalue is the data set's *principle component*. The eigenvectors are sorted by eigenvalue in descending order and the first  $p$  eigenvectors are selected forming the matrix  $\mathbf{E} = (\mathbf{e}_1, \dots, \mathbf{e}_p)$  [Smith02]. So only the  $p$  most significant eigenvectors are selected, assuming that the vectors with smaller variance encode noise only. By ignoring some eigenvectors of lesser significance (e.g. the vectors with eigenvalues not contained in 95% of the variance) information is lost. If the discarded eigenvectors are small the lost information can be neglected [Smith02]. The most significant or strongest eigenvectors with the largest eigenvalues represent the dimensions with the strongest correlation in the data set and are proportional to the covariance. In this way only those data set characteristics, that contribute most to the data set's variance, are kept. A simple example for the reduction of the dimensionality from  $n$ -D to 2-D is shown in Fig. 3.1. In this example two-dimensional data can be reduced to one-dimensional data by projecting onto the red line and using the resulting one-dimensional coefficients to describe the data. The orthogonal information, parallel to the green line, is lost. The red and green line represent the eigenvectors with the largest eigenvalues, which in turn represent the dimensions with the strongest correlation in the data set.

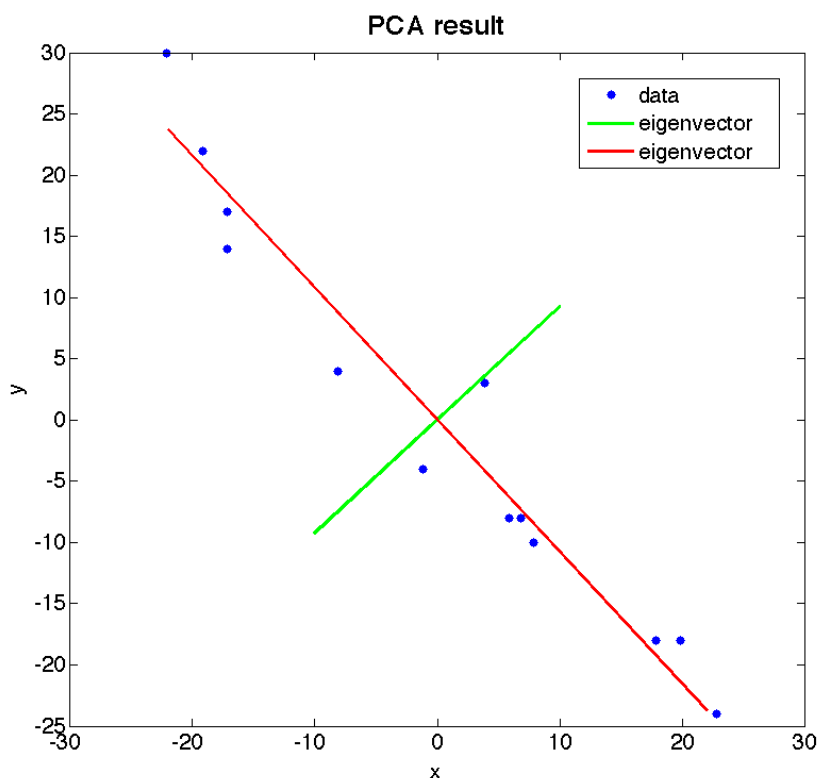


Figure 3.1.: PCA result where  $n$ -dimensions were reduced to 2-dimensions. The drawn in eigenvectors (red, green) with the largest eigenvalues represent the dimensions with the strongest correlation in the data set.

PCA is used in several areas of medical image processing including Shape Models [Taylor92], ASMs [Cootes92], AAMs [Cootes01] for the construction of point distribution-based shape models as presented in Sec. 3.3.

## 3.2. Minimum Description Length

The principle of MDL was first introduced in 1978 [Rissanen78]. It is based on the work of Solomonoff on the *Kolmogorov* or *algorithmic complexity*. MDL is a method for inductive inference and is related to *Bayesian Inference* and the *Minimum Message Length* (MML) principle.

The central idea of MDL is that on the one hand by compressing data every regularity in the data can be used and that on the other hand with finding regularities in data learning can be equated. In other words the goal is to learn about the data by compressing

it. Formally, by applying MDL it is tried to find the hypothesis or a combination of hypotheses in a given set of hypotheses  $\mathcal{H}$ , that compress a given data set  $D$  best.

MDL implements the *Occam's Razor* principle, of which the main statement is, that the most simple descriptions are the best. Therefore, if MDL has to choose between two models that fit the data equally well, it will choose the one that allows the shorter description of the data [Grunwald05].

To formulate a theoretically ideal MDL that suits the main idea of learning by compressing data, a data sequence and a description method to express the data properties are needed. A general choice for this description method could be a general-purpose computer language. Considering the *Kolmogorov Complexity*, which is the length of the shortest program that prints the sequence and then halts, it can be shown that the selection of the computer language does not matter, because the lengths of two programs written in different languages differ only by a constant, which in turn does not depend on the length of the data sequence as long as it is long enough. This fact is known as *invariance theorem*. Because of the uncomputability, arbitrariness and dependence on the chosen syntax the above described ideal MDL can not be used in practice [Grunwald05].

To transform an ideal MDL into a practical MDL, description methods that are less expressive than general-purpose computer languages have to be used. These description methods should fulfill two constraints. First they should always allow to compute the shortest descriptions length and second should be general enough to compress most of the regular sequences. The practical MDL can be formulated using two different approaches, the *crude* or *two-part code version* and the *refined* or *one-part code version* [Grunwald05].

The crude version of the MDL principle achieves the hypothesis  $H$  that best describes the data  $D$  by minimizing the sum  $L(H) + L(D|H)$ , where  $L(H)$  is the hypothesis' description length and  $L(D|H)$  is the length of the description of the data encoded with the hypothesis. The problem with crude MDL is that the description length of a hypothesis can differ largely when using different description methods (codes) and therefore the procedure is in danger to become arbitrary.

To overcome this problem the crude MDL was enhanced to the refined MDL. The crude version aims to find a single Hypothesis  $H \in \mathcal{H}$  for encoding. However the refined MDL aims at finding a full set  $\mathcal{H}$  of hypotheses for encoding the data. Another difference

is that according to its denotation the *one-part code* uses only a single one-part code with lengths  $\bar{L}(D|\mathcal{H})$  instead of two parts. This assures that whenever  $L(D|H)$  is small ( $\exists H \in \mathcal{H}$  that fits the data well),  $\bar{L}(D|\mathcal{H})$  will also be small. Refined MDL uses a second concept called *parametric complexity*  $\mathbf{COMP}(\mathcal{H})$ . It indicates a models ability to fit random data. Despite the improvements from crude to refined MDL, model selection using refined MDL is still a trade-off between a goodness of fit term  $L(D|\hat{H})$  ( $\hat{H}$  is the distribution in  $\mathcal{H}$ , that minimizes the code length) and the complexity term  $\mathbf{COMP}(\mathcal{H})$  [Grunwald05].

In [Thodberg03] the MDL principle was used to find point correspondences in training examples for ASMs and AAMs. This approach is used to reduce the number of shape pixels (represented as the shape's contour) of the used data sets resulting from the manual image annotation and to obtain a suitable ( $< 200$  landmarks per shape) amount of shape landmarks.

The  $N_S$  annotated shape contours  $\mathbf{c}_i$  ( $i = 1, \dots, N_S$ ), which are not of equal length, are aligned and corresponding points are calculated. These corresponding points, the so called landmarks, are located on the annotated shape contours. With these points the point position vectors  $\mathbf{v}_i$  (see Eq. 3.3) for each shape are generated, that are used for the later construction of the Shape Model (see Sec. 3.3). Each shape contains the same number of landmarks and each landmark always describes the same anatomical position.

### 3.3. Shape Models

Objects in images can be represented using statistical models of the objects' shape. A Point Distribution Model (PDM) [Taylor92] constructs a shape model by computing the significant eigenmodes of a shape population, e.g. assuming a multivariate Gaussian distribution of the shape parameters.

These shapes consist of a set of  $n$  points or landmarks. The points can have any dimension, but throughout this paper they are considered to be 2-dimensional. Before modeling the non-rigid shape variation the shapes are normalized for similarity transformations, i. e. translation, rotation and scaling (the *pose*). Thus the parameters to define a unique shape in an image are shape and pose parameters. Considering these facts,

shape can be defined as the quality of a set of points that is invariant to transformations affecting pose.

Using this shape information the goal is to build models which provide the abilities to represent shapes and to generalize to new shapes within the distribution of the shapes in a predefined training set.

The first step to create a training set for shape model generation is to define landmarks for several objects. To obtain landmarks for a shape usually a human expert annotates several images containing the corresponding object. Landmarks are derived from this manually annotated image contours by applying MDL [Thodberg03]. Then a vector  $\mathbf{v}_i$  for all  $i \in 1, \dots, N_S$  annotated shapes for  $d$ -dimensional landmarks is defined as

$$\mathbf{v}_i = (l_{1_1}, \dots, l_{1_d}, l_{2_1}, \dots, l_{2_d}, \dots, l_{n_1}, \dots, l_{n_d}) \quad (3.3)$$

The training set is then aligned using Procrustes Analysis, which minimizes  $\sum |\mathbf{v}_i - \bar{\mathbf{v}}|^2$ , where  $\mathbf{v}_i$  is the  $i^{\text{th}}$  point position vector and  $\bar{\mathbf{v}}$  is the mean of all vectors, i. e. the mean shape.

To be able to generate new shapes out of the training set a parameterized model  $\mathbf{v} = \mathbf{M}(\mathbf{c})$ ,  $\mathbf{c} = (b, t)$  of the distribution of the  $N_S$  point position vectors  $\mathbf{v}_i$  is defined.  $\mathbf{b}$  is a vector containing the model parameters (Eq. 3.5) and  $\mathbf{t}$  encodes the pose parameters (see below). With the help of this model it is possible to generate new shapes  $\mathbf{v}$  and to estimate the distribution  $p(\mathbf{v})$  of these new vectors.

The actual point distribution model is built by applying PCA on  $\mathbf{V} = (\mathbf{v}_1, \dots, \mathbf{v}_{N_S})$ , yielding eigenvectors  $\mathbf{e}_1, \dots, \mathbf{e}_e$  with  $e = \min(nd, N_S)$ . Using  $e^* < e$  eigenvectors (thereby neglecting the modes with small variance, which are considered to model noise only) any shape  $\mathbf{v}$  within the subspace spanned by the training set can be represented by:

$$\mathbf{v} \approx \bar{\mathbf{v}} + \mathbf{E}\mathbf{b} \quad (3.4)$$

where  $\mathbf{E} = (\mathbf{e}_1, \dots, \mathbf{e}_{e^*})$  is the basis of the eigenspace and  $\mathbf{b}$  is a vector of length  $e^*$  which defines the parameters for the deformable model [Cootes01]:

$$\mathbf{b} = \mathbf{E}^T(\mathbf{v} - \bar{\mathbf{v}}) \quad (3.5)$$

The resulting model  $\mathbf{E}$  represents the shape variation of the modeled objects utilizing a single parameter vector  $\mathbf{b}$ . Each element of  $\mathbf{b}$  controls one mode of shape variation, with



the first modes being responsible for the highest variation, in descending order.

To ensure that the generated shapes are similar to those in the training set the parameter vectors are always limited to the range  $\pm 3\sqrt{\lambda_i}$ , where  $\lambda_i$  is the eigenvalue and also the variance of the  $i^{\text{th}}$  parameter  $b_i$  in the training set.

In addition to these modes the transformations translation, scaling and rotation need to be taken into account. Therefore a new linear parameter vector  $\mathbf{t} = (s_x, s_y, t_x, t_y)^T$  is introduced, controlling rotation  $\theta$ , scaling  $s$  and translation  $(t_x, t_y)$ , with  $s_x = s \cos \theta - 1$  and  $s_y = s \sin \theta$ .

Combining the parameter vector of the PCA  $\mathbf{b}$  and the parameter vector for translation, rotation and scaling  $\mathbf{t}$  results in the combined parameter vector

$$\mathbf{c} = (\mathbf{b}^T, \mathbf{t}^T)^T. \quad (3.6)$$

The first three eigenmodes for the metacarpal bones data set are shown in Fig. 3.2. The eigenmodes and eigenvectors of the shape models generated for all used data sets can be found in the Appendix, Ch. A.1.

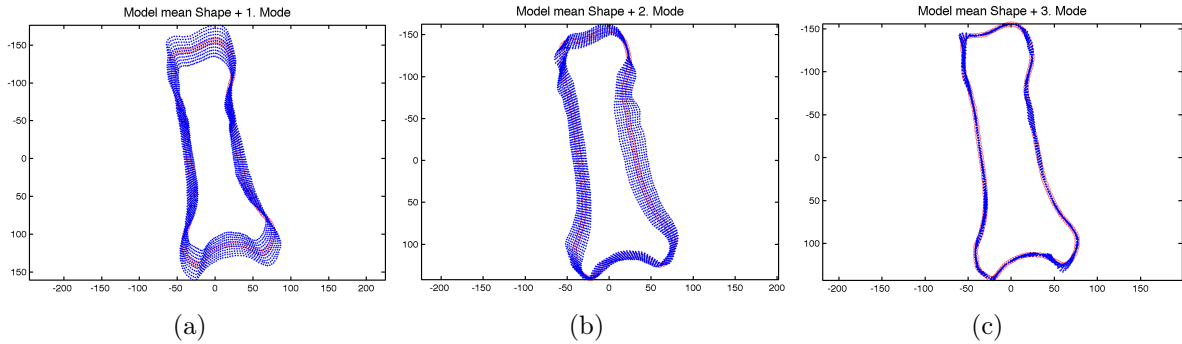


Figure 3.2.: Visualization of the first three calculated eigenmodes for the metacarpal bones data set. The mean shape is drawn in red, the modes are blue.

# Chapter 4.

## Shape Particle Filter

This chapter concentrates on the Shape Particle Filter approach proposed in [deBruijne04a, deBruijne04b]. Sec. 4.1 describes the basic techniques used during the filtering process, namely Gabor Jets and the  $k$ -Nearest Neighbors ( $k$ -NN) classifier. The filtering pipeline is divided into two major parts, *training* (see Sec. 4.2) and *search* (see Sec. 4.3).

During training first the mean shape is derived from hand annotated training images. Then a region map is manually defined on this mean shape, representing regions which are presumed to be distinct and of significance to the segmentation process.

Local image descriptors (e.g. Gaussian derivatives, Gabor filters [Yoshimura00], see Sec. 4.1.1) are computed for all training images. The region map is warped back from the mean shape onto each training shape according to the manual annotation using Thin Plate Spline warping (TPS) [Bookstein89]. Having corresponding regions marked in each training image a distribution of the corresponding descriptors/features for each region can be estimated by sampling from the image. The shape model, the region map and the feature descriptors learnt for the regions thus constitute the prior knowledge of Shape Particle Filters.

During search on a test image, a  $k$ -NN classifier (see Sec. 4.1.2) is used to classify the image's pixels  $j \in 1 \dots N_P$  resulting in one region probability map  $P_l$  per region  $l$  (see Eq. 4.7). The actual segmentation step uses these probabilities to optimize a fitness function encoding the belief in the segmentation corresponding to a given shape model parameter vector.

The entire Shape Particle Filtering process pipeline is shown in Fig. 4.1.

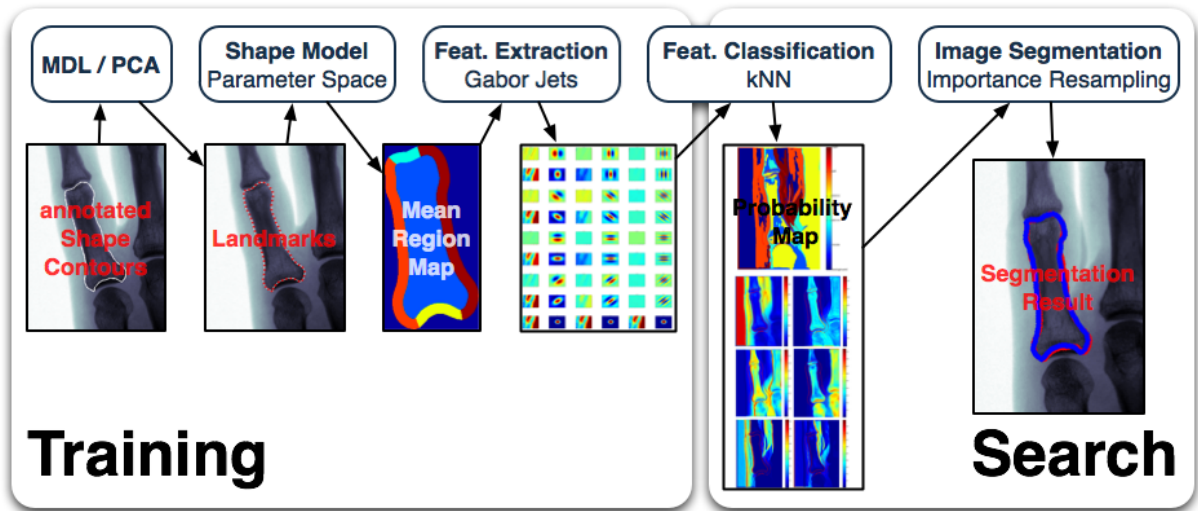


Figure 4.1.: Flowchart of a Shape Particle Filter. Landmarks are obtained from manually annotated contours, forming the basis for the shape model. A region map based on the shape model’s mean shape is generated that in turn is used to determine image region features. Test images are then classified yielding probability maps that are used in the final image segmentation step to estimate the confidence in each hypothesis.

## 4.1. Preliminaries

In this section the feature extraction method as well as the applied classifier that are both part of the Shape Particle Filter pipeline are presented. As proposed in [deBruijne04a, deBruijne04b] Gabor Jets are used as image features. Alternative feature extraction algorithms are, e. g., histograms, gradient, amplitude or phase information as well as the Monogenic Signal.

Similar to [deBruijne04a, deBruijne04b] a  $k$ -NN Classifier was set up for the classification step. Other possible classifiers that can be applied in this step are  $k$ -NN with a  $kd$ -Tree search, Support Vector Machines and Random Forests. A comparison of the performance of a  $k$ -NN Classifier, a  $k$ -NN with a  $kd$ -Tree search and a Support Vector Machine is provided in Sec. 5.3.

### 4.1.1. Gabor Jets

Gabor Jets are the output from a set of Gabor filters. The advantage of Gabor Jets lies in their phase invariance to edge positions, if the filter responses are converted to amplitude and phase [Yoshimura00].

A Gabor filter consists of two functions, a complex sinusoidal called *carrier* and a Gaussian-shaped function known as the *envelope*

$$g(x,y) = s(x,y)w_r(x,y) \quad (4.1)$$

where  $s(x,y)$  is the carrier and  $w_r(x,y)$  is the envelop [Movellan96]. The complex sinusoidal carrier function is defined as

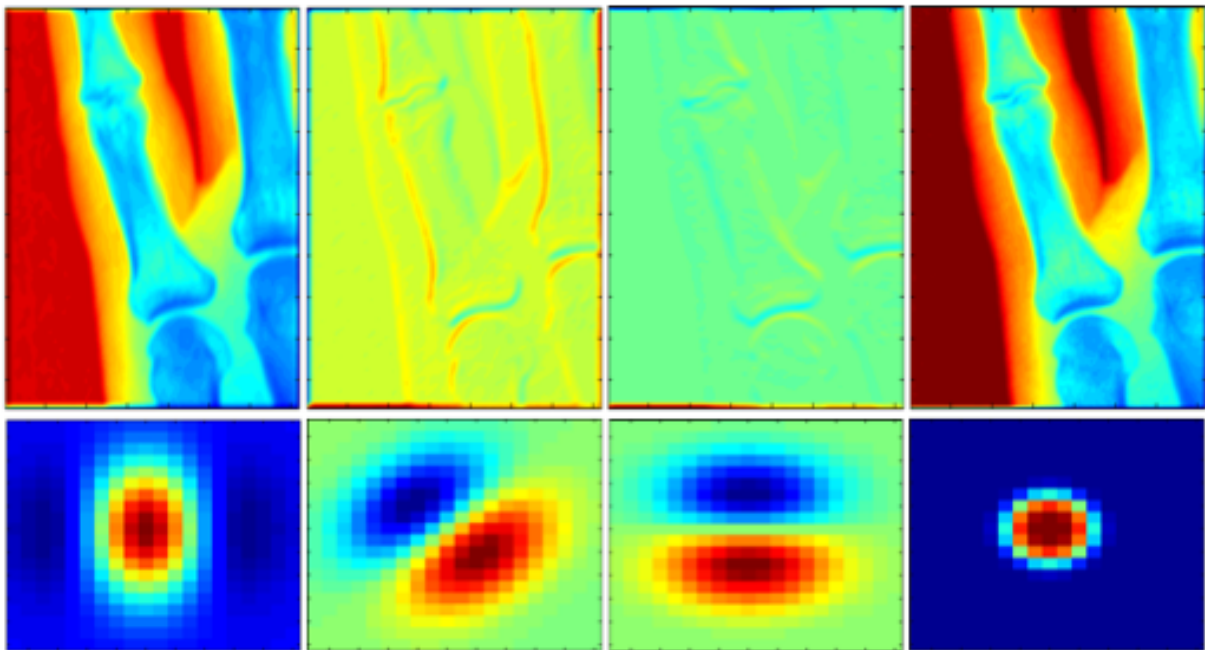
$$s(x,y) = e^{j(2\pi(u_0x+v_0y)+P)} \quad (4.2)$$

where  $u_0$  and  $v_0$  define the spatial frequency in Cartesian coordinates and  $P$  the phase [Movellan96]. The Gaussian envelop can be written as

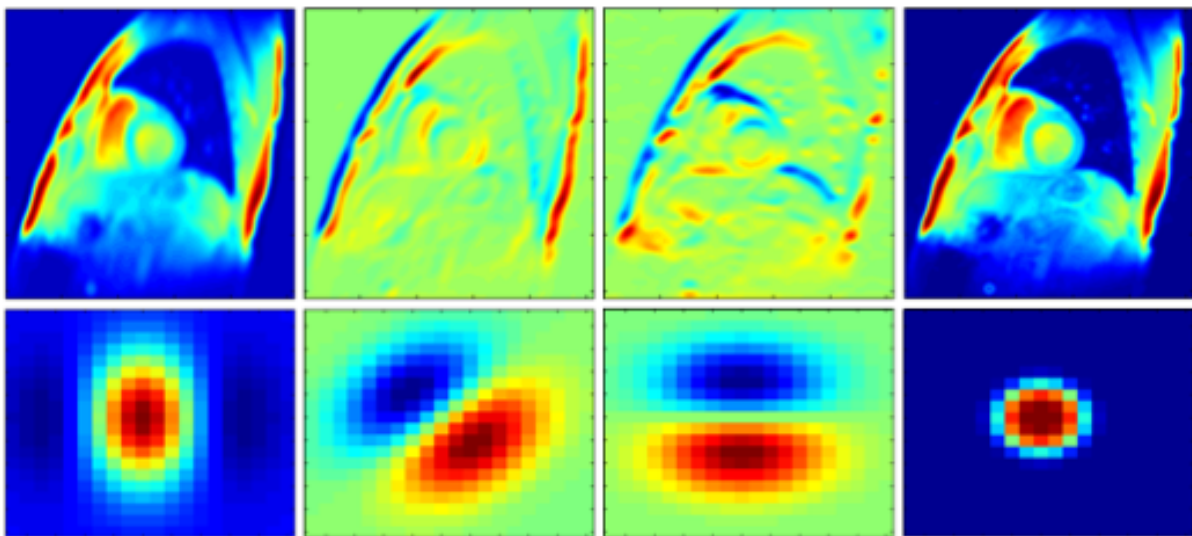
$$w_r = Ke^{-\pi(a^2(x-x_0)_{r(\theta)}^2 + b^2(y-y_0)_{r(\theta)}^2)} \quad (4.3)$$

where  $K$  is a scale factor for the envelop's magnitude,  $(x_0, y_0)$  is the location of the peak of the Gaussian envelop,  $a$  and  $b$  are scaling parameters for the two axis of the Gaussian and  $r(\theta)$  stands for a rotation operation, describing a clockwise rotation with the angle  $\theta$  [Movellan96]. In the implementation used in the scope of this thesis the two scaling parameters are equal (i. e.  $a = b$ ).

The Gabor filters and extracted features are shown in Fig. 4.2(a) for the metacarpal bones and in Fig. 4.2(b) for the hearts.



(a)



(b)

Figure 4.2.: Example images of 4 extracted Gabor features and the according Gabor filter for the metacarpal bones data set (a) and the hearts data set (b).

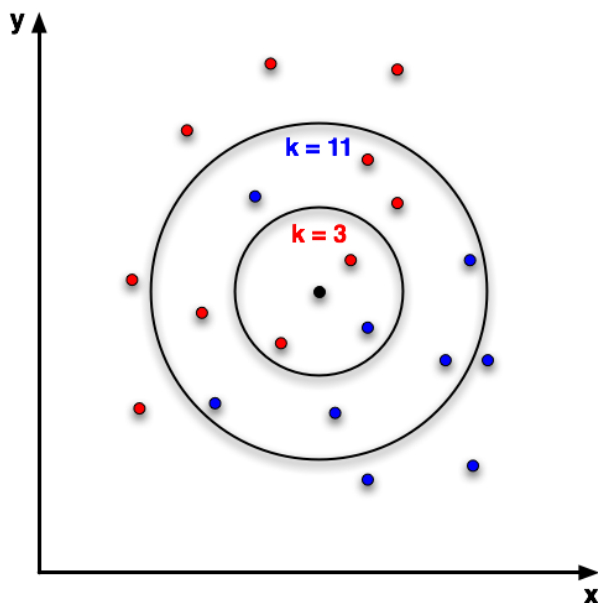


Figure 4.3.: A simple 2D classification example using the  $k$ -Nearest Neighbors algorithm.

The black point in the center is classified either as class *red* or class *blue*, depending on the selected value of  $k$ .

#### 4.1.2. $k$ -Nearest Neighbors

The nearest neighbor algorithm was first formulated by Fix and Hodges ([Fix52, Fix89]) and they investigated the  $k$ -nearest neighbor ( $k$ -NN) rule [Cover67]. The  $k$ -NN algorithm simply classifies a pixel by assigning it to the class that is most common amongst its  $k$  nearest neighbors.  $k$  is usually chosen as a small, odd integer. To calculate the actual distance between the pixels for example euclidean distance can be used. A two-dimensional example is visualized in Fig.4.3. To classify the *black* data point, the distances to all surrounding points are calculated. If  $k = 3$ , the point is classified as class *red* (two out of three neighboring points belong to class *red*). On the other hand if  $k = 11$  the classification result is class *blue* (six out of 11 neighboring points belong to class *blue*).

## 4.2. Training

During training the Shape Particle Filter algorithm learns the local shape and appearance of the object of interest on a given set of  $N_S$  training images. The shape information is incorporated in a shape model describing the landmarks extracted from manually annotated shape contours forming the shape vectors  $\mathbf{v}_i$  ( $i \in 1, \dots, N_S$ ). Based on the mean shape  $\bar{\mathbf{v}}$  a so called mean region map  $\bar{\mathbf{R}}$  (also named *class template* or *labeling map*) of distinct regions is defined.

A region map  $\bar{\mathbf{R}}$  is created by simply assigning a region or label  $l \in 1, \dots, N_L$  to each pixel  $j$  in a predefined area of interest.

$$\bar{\mathbf{R}}(j) = l \quad j \in 1 \dots N_P, \quad l \in 1 \dots N_L \quad (4.4)$$

where  $N_P = N_J * N_L$  and  $N_J$  denotes the number of pixels per region  $l$ .

The number of regions  $N_L$  consists of the number of regions inside the shape  $N_{inner}$  and the number of regions around the shape  $N_{border}$ , e. g.  $N_L = N_{inner} + N_{border}$ . The width of the outside region has to be empirically chosen.

Furthermore it is possible to build the region map not only from one structure, but to use several neighboring structures, e. g. taking not only one lumbar vertebra into account, but also its neighboring vertebrae [deBruijne04a, deBruijne04b]. Using this approach allows to incorporate the underlying information of nearby distinctive structures which reduces the possibility that the algorithm converges to otherwise similar objects in the image. On the other hand region maps of complex compound structures strongly constrain the segmentation algorithms generalization abilities, e. g. a single object within a group described with a region map including all this objects.

The definition of the region map in the scope of the original Shape Particle Filtering scheme is done based on human intuition, i. e. *manually*.

**Manually Predefined Region Map** As described above the region map is based on the medical object of interest's shape representation. The easiest and most intuitive way to divide an object into several regions is to mark the inside and the outside as two different regions. Considering the fact that in the area around the shape all the information about neighboring structures, e. g. changes in tissue, bones, organs, etc. is

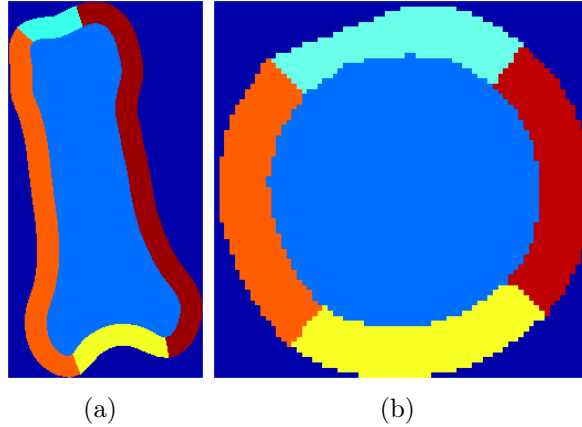


Figure 4.4.: Examples of two manually predefined region maps with five regions (inside, above, below, left, right) for the metacarpal bones (a) and the hearts (b) data sets.

encoded, a further partition of the outer region is conducted. In this implementation the outer region is therefore created to roughly represent four sides of the shape, e.g. left, right, above and below and therefore  $N_{border} = 4$ . This region partition was chosen, because the four regions have similar appearance, i.e. contain similar features. The inside of the shape is defined as one region with  $N_{inside} = 1$  resulting in an overall number of regions  $N_L = 5$ . Examples of these manually defined regions can be seen in Fig. 4.4(a) and 4.4(b).

After creating the mean region map  $\bar{\mathbf{R}}$ , it is warped back to each training shape using TPS warping, so that for each region a distribution of the corresponding descriptors/features can be estimated by sampling. This results in  $i = 1, \dots, N_S$  region maps  $\mathbf{R}_i$ , e.g. one region map for each training shape.

**Image Feature Extraction** To obtain information about the appearance of the processed image and the objects contained, feature descriptors have to be extracted. In the Shape Particle Filter implementation described in this chapter Gabor Jets were used for the feature extraction process (see Sec. 4.1.1) forming the feature matrix

$$\mathbf{F} = [\mathbf{f}_1, \dots, \mathbf{f}_{N_p}]^T \quad (4.5)$$

where  $\mathbf{f}_j$  are the feature vectors for each pixel  $j = 1, \dots, N_p$  in the image. The amount of extracted features and therefore the size of all feature vectors  $\mathbf{f}_j$  depends on the chosen



number of rotation angles and the number of frequencies.

The combination of the extracted features, the shape model and the previously built region maps forms the prior knowledge of Shape Particle Filters and therefore the basis for the later image classification.

### 4.3. Search

During the search part the actual segmentation based on the previously learned features, the region maps and shape model is conducted.

**Image Feature Classification** Using the extracted features and the class label information provided by the region maps of a set of training images, a moderated  $k$ -NN classifier [deBruijne04a, deBruijne04b] is used to classify the pixels of the test image features  $\mathbf{F}$ , yielding the probability of a pixel  $j$  belonging to a region  $l$

$$\gamma(j|\mathbf{F}, \bar{\mathbf{R}}) = \frac{k_l + 1}{k + N_L} \quad (4.6)$$

where  $k_l$  among the  $k$  nearest neighbors belong to region  $l$  and  $N_L$  is the number of regions in the mean region map  $\bar{\mathbf{R}}$ . This ensures that probabilities for finite  $k$  are always nonzero. For each pixel  $j$  of the test image the probability of belonging to a specific region is calculated using the classification results. This results in an image-sized probability map  $P_l$  for each region  $l \in 1, \dots, N_L$  of the region map, e. g. a map of the same size as the test image with the probabilities for each pixel  $j \in 1, \dots, n$  to belong to the region  $l$ .

$$P_l = \gamma(j|\mathbf{F}, \bar{\mathbf{R}}) \quad (4.7)$$

where  $j \in 1, \dots, n$  are the pixel in the test image,  $l$  is the region map label and  $\gamma(j|\mathbf{F}, \bar{\mathbf{R}})$  is the probability of pixel  $j$  belonging to region  $l$  resulting from the trained  $k$ -NN classifier.

**Segmentation** The optimization is performed using Particle Filtering by importance resampling processes [deBruijne04a, deBruijne04b], estimating the posterior distribution of the shapes given the image by means of the fitness function in Eq. 4.8. Initially, a random set of shapes, the particles, represented by shape parameter vectors  $\mathbf{c}_i$  are generated according to the distribution of the prior shape model.

The region map is warped according to the shape defined by the shape parameters and for each region the cumulative probability is computed by summing up the probabilities from the corresponding probability map.

By this, a weight can be applied to each particle considering their likelihood,

$$\pi(\mathbf{c}_i) = \frac{\sum_{l=1}^{N_L} \sum_{j=1}^{N_P} P_l^l(j)}{N_L}, \quad (4.8)$$

where  $N_L$  is the number of regions and  $P_l$  is the map containing the probability for each pixel  $j \in 1, \dots, N_P$  belonging to region  $l$ . New particles are generated from the current set of particles by weighting them with their likelihood  $\pi(\mathbf{c}_i)$  and randomly sampling in parameter space around these particles with probabilities proportional to the weights. While this resampling process is repeated the initial sparse particles evolve into a distribution with high density around the most likely shapes.

In the scope of this thesis a Differential Evolution based segmentation approach is applied in contrast to the original proposed importance resampling, concentrating on the extracted features and the optimized region maps. Therefore no comparison to the original implementation is performed.

# Chapter 5.

## Optimizing Shape Particle Filters by Automated Training

In this chapter the thesis' main contributions are explained. It is structured similar to Chapter 4. Sec. 5.1 describes the basic techniques used during the filtering process, namely the Monogenic Signal, the compared classification optimizations, *kd*-Trees and Support Vector Machines and the Differential Evolution stochastic optimization framework. Again, the filtering pipeline is subdivided into two major parts, *training* (see Sec. 5.2) and *search* (see Sec. 5.3).

The previously described Shape Particle Filter scheme is modified to increase computational performance and accuracy. The following modifications are conducted:

- Replacing the manual definition of the region map by an automatic derivation of the number and location of optimal regions.
- Introducing a per-pixel region map, that does not require the time consuming image feature classification step.
- Replacing the previous used Gabor Jets with a more sophisticated features based on the Monogenic Signal.

The first step of the pipeline, namely the derivation of the shape model, remains unchanged. In the next step instead of Gabor Jets, the Monogenic Signal is used for the image feature extraction. Concerning the region map generation two enhancements are presented: an automatic region map and a per-pixel region map. Depending on the kind of region map that is used during the segmentation two different subsequent operational sequences arise.

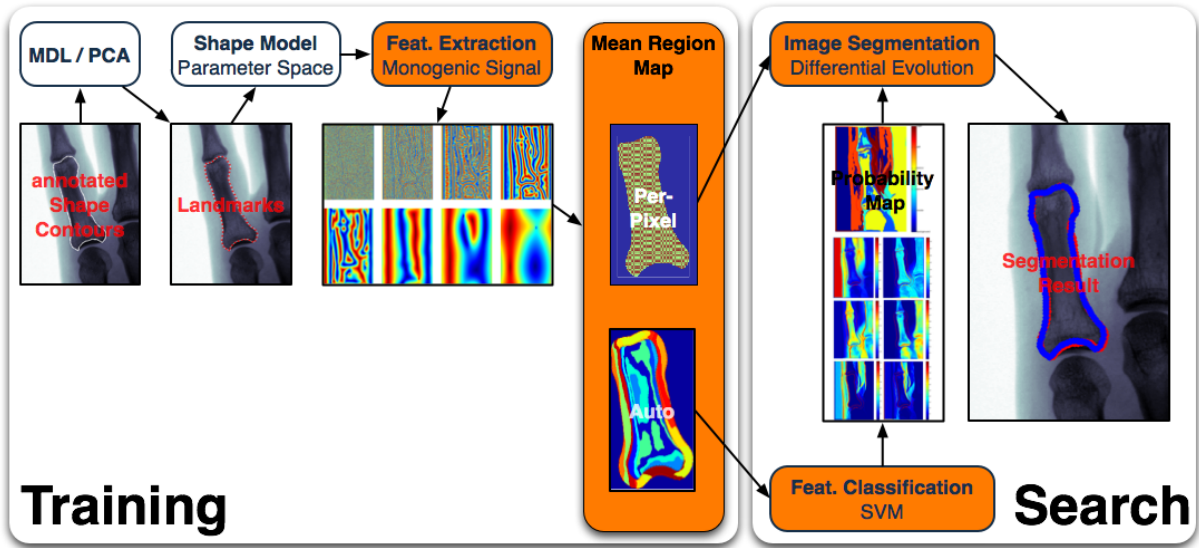


Figure 5.1.: Flowchart of the proposed improved Shape Particle Filter. Landmarks are obtained from manually annotated contours, forming the basis for the generated Shape Model. A region map based on the shape models mean shape is generated that in turn is used to determine image region features. Two possible region map improvements are implemented: an automatic region map and a per-pixel region map. Using an automatic region map, in the next step the test images are classified yielding probability maps that are used in the final image segmentation step. This step is unnecessary when using a per-pixel region map. The steps in the flowchart differing to the original Shape Particle Filter implementation are highlighted orange.

When using an automatically derived region map a following classification step is needed to construct the probability maps that in turn are needed for the Differential Evolution based image segmentation, e.g. the calculation of the weight or residual.

Using a per-pixel region map the classification step can be omitted, leading to a significant increase in computational performance but still sustaining or even improving the segmentation accuracy. Instead of a probability map appearance information obtained by using the Monogenic Signal is directly used to compute the residual for the Differential Evolution.

The shape segmentation is performed similar to the original Shape Particle Filter implementation. The actual Particle Filtering is done by using a Differential Evolution based approach [Braak06], which is described in Sec. 5.3.

The improved Shape Particle Filtering process pipeline is shown in Fig. 5.1.

## 5.1. Preliminaries

In this section the necessary theoretical background for the later introduced applications are described. In Sec. 5.1.1 the Monogenic Signal, that is used for the image feature extraction, is presented. The description of  $kd$ -Trees (Sec. 5.1.2) and Support Vector Machines (Sec. 5.1.3), as well as the  $k$ -NN (Sec. 4.1.2) form the basis for the classification speed comparison presented in Sec. 5.3. The actual segmentation is achieved by applying Differential Evolution described in Sec. 5.1.4.

### 5.1.1. Monogenic Signal

The Monogenic Signal was introduced in [Felsberg01] and is a generalization of the 1D analytic signal to two dimensions. By convolving a 1D signal  $f(x)$  with its Hilbert transform  $f_H(x = h(x) * f(x))$  the local amplitude

$$\begin{aligned} A(x) &= \|f_A(x)\| \\ &= \sqrt{f^2(x) + f_H^2(x)} \end{aligned} \quad (5.1)$$

and local phase ( $\varphi(x) \in [-\pi, \pi)$ )

$$\begin{aligned} \varphi(x) &= \arg(f_A(x)) \\ &= \arctan2(f_H(x), f(x)) \end{aligned} \quad (5.2)$$

can be derived from the analytic signal  $f_A(x) = f(x) + f_H(x)$ .

The Hilbert transform, that is usually applied to obtain a 2D analytical signal, is replaced by the Riesz transform, which in turn is a 2D generalization of the Hilbert transform. Combining a 2D signal (or image)  $f(x_1, x_2)$  and the Riesz transformed signal represented by  $h_1(x_1, x_2)$  and  $h_2(x_1, x_2)$  results in the Monogenic Signal

$$f_M(x_1, x_2) = (f(x_1, x_2), (h_1 * f)(x_1, x_2), (h_2 * f)(x_1, x_2)), \quad (5.3)$$

where

$$\begin{aligned} H_1(u_1, u_2) &= \frac{u_1}{|u|}, H_2(u_1, u_2) \\ &= \frac{u_2}{|u|}. \end{aligned} \quad (5.4)$$

Using these equations the local amplitude  $A_f$  (Eq. 5.5), local phase  $\varphi$  (Eq. 5.6) and local orientation  $\theta$  (Eq. 5.7) can be extracted.

$$A_f(x_1, x_2) = \sqrt{f^2 + (h_1 * f)^2 + (h_2 * f)^2} \quad (5.5)$$

$$\varphi(x_1, x_2) = \text{acos}\left(\frac{f(x_1, x_2)}{A_f(x_1, x_2)}\right), \varphi \in [0, \pi] \quad (5.6)$$

$$\theta(x_1, x_2) = \text{atan2}(h_2 * f, h_1 * f), \theta \in [-\pi, \pi] \quad (5.7)$$

Examples of features extracted using the Monogenic Signal are shown in Fig. 5.9.

### 5.1.2. kd-Trees

A *kd*-Tree is a data structure for storing data in *k*-dimensional spaces that then can be used for associative searches. It is an extension of a binary tree where every node represents a *k*-dimensional point. Each node  $P$  contains *k* keys  $K_n(P)$  ( $n = 0, \dots, k - 1$ ) containing the data and two pointers. These pointers are either null or point to another node in the *kd*-Tree and therefore each pointer can be seen as a subtree forming left and right branches of the node. For keys located on a left branch of node  $P$  in depth  $j$  the following condition is true

$$K_j(\text{LEFT}) < K_j(P). \quad (5.8)$$

Likewise for keys located on a right branch of node  $P$  in depth  $j$

$$K_j(\text{RIGHT}) > K_j(P) \quad (5.9)$$

is true. A special case occurs, if the keys are equal. In this case the decision where the key belongs must be based on the remaining keys. This can be done by tracking back the branch and examining the keys there, or by defining a superkey, that in case

of equality returns a predefined value. But actually the decision what to do is arbitrary, depending on the respective area of application.

A 2D tree decomposition is shown in Fig. 5.2 and its graph representation in Fig. 5.3.

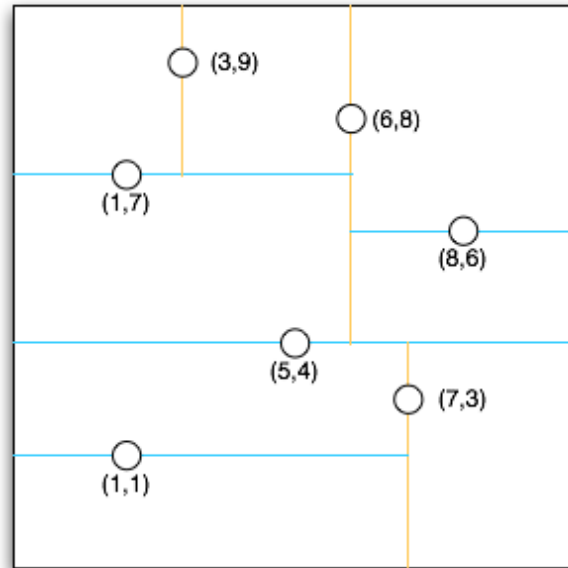


Figure 5.2.: A 2D tree decomposition.

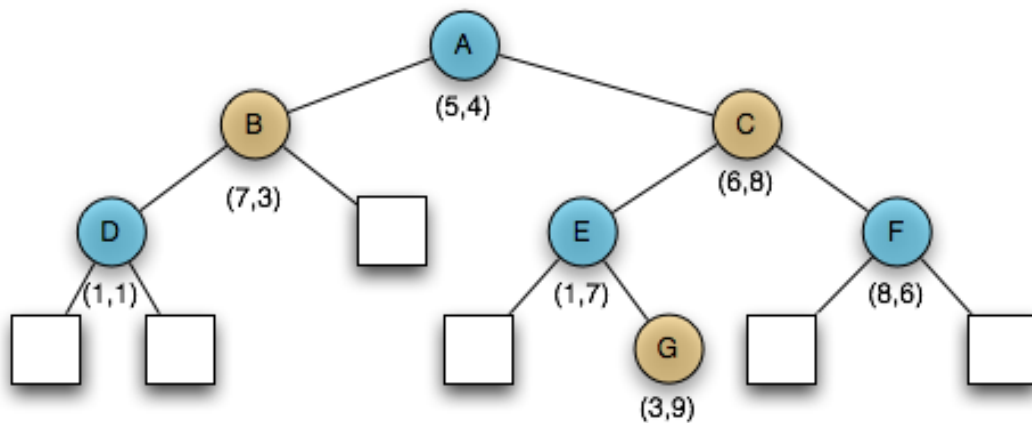


Figure 5.3.: Graph representation of a 2D tree. The squares represent null pointers.

### 5.1.3. Support Vector Machines

Based on statistical learning theory [Vapnik95] and originally developed for pattern recognition Support Vector Machines (SVM) are supervised learning methods that are used for solving classification and regression problems. By applying an a priori chosen nonlinear mapping (the *kernel*), an input vector is mapped into a high dimensional feature space and an optimal separating hyperplane is created there. To achieve this, a generalization of the SVM in the high-dimensional space has to be considered.

SVMs can be divided into linear SVMs and nonlinear SVMs according to the kind of the decision function (kernel) used. Furthermore two cases have to be distinguished, SVMs used for classification of linearly separable data and SVMs used for non-separable data. In this short description of the used SVM the focus lies on linear SVMs. A detailed description of SVMs, especially of the nonlinear form and the kernels used therein can be found in [Vapnik95, Burges98, Schölkopf00, Bishop07].

Consider a linear SVM trained on linearly separable data represented by  $n$  input vectors  $\mathbf{x}_1, \dots, \mathbf{x}_n$ . A label  $t_i$  is assigned to each data vector  $\mathbf{x}_i$ , resulting in targets or labels  $t$ , where  $t_i \in \{-1, 1\}$ . A new data vector  $\mathbf{x}$  is classified by the sign of its according label  $t(\mathbf{x})$ . We assume a separating hyperplane  $H$ , where points which lie on  $H$  satisfy  $\mathbf{w}\mathbf{x} + b = 0$  ( $\mathbf{w} \perp H$ ,  $|b|/\|\mathbf{w}\|$  perpendicular distance from  $H$  to origin). SVMs try to find the classification solution with the smallest generalization error. This is done by choosing the decision boundary so that the *margin* is maximized. The margin is defined as the smallest distance between the decision boundary and all the samples.

Defining two new hyperplanes  $H_1 : \mathbf{x}_i\mathbf{w} + b = 1$  ( $\mathbf{w} \perp H$ ,  $|1 - b|/\|\mathbf{w}\|$ ) and  $H_2 : \mathbf{x}_i\mathbf{w} + b = -1$  ( $\mathbf{w} \perp H$ ,  $|-1 - b|/\|\mathbf{w}\|$ ), data points that satisfy Eq. 5.10 lie on  $H_1$  and data points satisfying Eq. 5.11 lie on  $H_2$ .  $H_1$  and  $H_2$  have the same normal  $\mathbf{w}$  and are therefore parallel.

$$\mathbf{x}_i\mathbf{w} + b \geq +1 \quad \text{for } t_i = +1 \quad (5.10)$$

$$\mathbf{x}_i\mathbf{w} + b \leq -1 \quad \text{for } t_i = -1 \quad (5.11)$$

Combining these inequalities yields to the following set of inequalities

$$t_i(\mathbf{x}_i\mathbf{w} + b) - 1 \geq 0 \quad i = 1, \dots, n \quad (5.12)$$

Furthermore no training points fall between them. The training points lying on one of



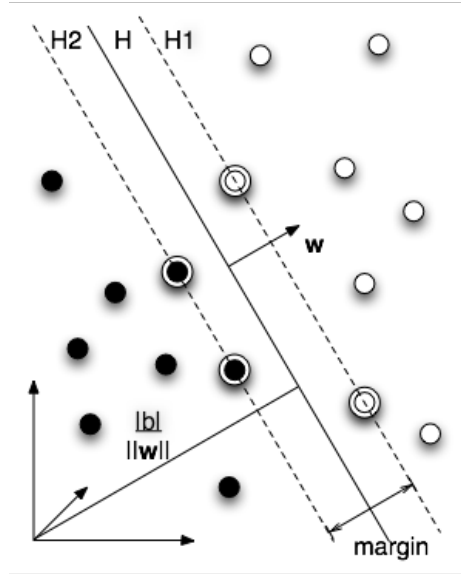


Figure 5.4.: The separating hyperplanes for a linear separable 2D classification problem. The support vectors are marked with circles.

the two hyperplanes  $H_1$  or  $H_2$  and hence satisfying Eq. 5.12 are called *support vectors*. Removing any of them would change the classification result.

The distances between  $H$  and the two other hyperplanes are  $d_1 = d_2 = 1/\|\mathbf{w}\|$  and therefore the margin is  $2/\|\mathbf{w}\|$ .

The solution for a simple 2D classification problem is shown in Fig. 5.4.

By reformulating the problem into a Lagrangian a later generalization for nonlinear data classification is possible. Therefore unconstrained positive Lagrange multipliers  $\alpha_i$ ,  $i = 1, \dots, n$  are multiplied with the constraint equations and the result is subtracted from the objective function, leading to the Lagrangian

$$L_P \equiv \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n (\alpha_i t_i (\mathbf{x}_i \mathbf{w} + b)) + \sum_{i=1}^n (\alpha_i). \quad (5.13)$$

Minimizing  $L_P$  leads to a convex quadratic programming problem. A dual formulation of the problem called the *Wolfe dual* can be used to get the dual formulation  $L_D$  of the Lagrangian  $L_P$ , as follows

$$L_D = \sum_i (\alpha_i) - \frac{1}{2} \sum_{i,j} (\alpha_i \alpha_j t_i t_j \mathbf{x}_i \mathbf{x}_j) \quad (5.14)$$

under the conditions

$$\mathbf{w} = \sum_i (\alpha_i t_i \mathbf{x}_i) \quad (5.15)$$

$$\sum_i (\alpha_i t_i) = 0. \quad (5.16)$$

SVM training is then done by maximizing  $L_D$  with respect to  $\alpha_i$ . The exact description of the derivation of  $L_D$  can be found in [Burges98].

Applying this algorithm on non-separable data will lead to no feasible solution (an arbitrary large growing objective function i.e. the dual Lagrangian). This can be overcome by extending the constraints in Eq. 5.10 and Eq. 5.11 with further costs in the form of positive slack variables  $\xi_i$ ,  $i = 1, \dots, n$  to

$$\mathbf{x}_i \mathbf{w} + b \geq +1 - \xi_i \quad \text{for } t_i = +1 \quad (5.17)$$

$$\mathbf{x}_i \mathbf{w} + b \leq -1 + \xi_i \quad \text{for } t_i = -1 \quad (5.18)$$

$$\xi_i \geq 0 \quad \forall i. \quad (5.19)$$

The sum over all slack variables  $\xi_i$  forms an upper bound on the number of training errors. The dual formulation  $L_D$  then becomes

$$L_D \equiv \sum_i (\alpha_i) - \frac{1}{2} \sum_{i,j} (\alpha_i \alpha_j t_i t_j \mathbf{x}_i \mathbf{x}_j) \quad (5.20)$$

with the conditions

$$0 \leq \alpha_i \leq C \quad (5.21)$$

$$\sum_i (\alpha_i t_i) = 0 \quad (5.22)$$

$$\mathbf{w} = \sum_i^{n_s} (\alpha_i t_i \mathbf{x}_i) \quad (5.23)$$

where  $n_s$  represents the number of support vectors and  $C$  is a manually defined upper bound. By comparing the linear separable and non-separable case it is obvious that the two dual Lagrangians differ only in their constraint conditions, more precisely in the upper bound  $C$  of the Lagrangian multipliers in the non-separable case. The separating hyperplanes for the non-separable case are shown in Fig. 5.5. To transform the primal Lagrangian  $L_P$  the Karush-Kuhn-Tucker conditions are needed. Details hereunto can be

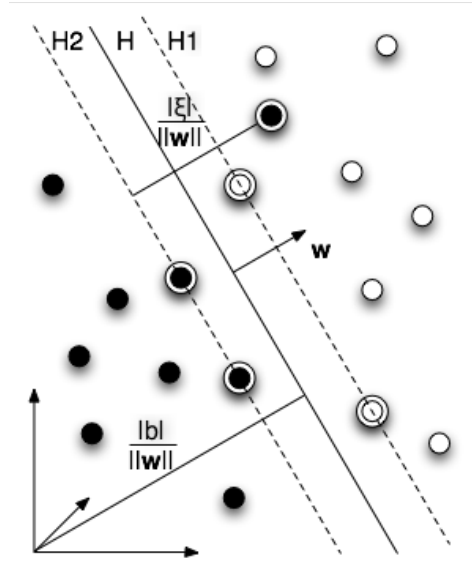


Figure 5.5.: The separating hyperplanes for a linear non-separable 2D classification problem. The support vectors are marked with circles.

found in [Burges98].

#### 5.1.4. Differential Evolution

Differential Evolution (DE) is a genetic algorithm and aims at optimizing functions based on populations in parameter space, which in this case is the subspace (restricted to plausible models) of the model parameters  $\mathbf{c}_i$ .

DE is a parallel direct search method that uses  $N_c$   $d$ -dimensional (in the scope of this thesis  $d = 2$  or  $d = 3$  depending on the used shapes dimension) parameter vectors  $\mathbf{c}_i$  ( $i = 1, \dots, N_x$ ) (i. e.  $N_c$   $d$ -dimensional Markov chains [Braak06], see Fig. 2.1) as members of a population  $\mathbf{C}_g$  for each generation  $g$ .

Starting with parameter vectors randomly drawn from the training distribution at generation  $g = 1$ , during each generation  $g + 1$ ,  $N_c$  new parameter vectors / shape hypotheses  $\mathbf{c}^h$  are then generated by adding the weighted difference vector between two population members to a third member

$$\mathbf{c}^h = \mathbf{c}_1 + \gamma(\mathbf{c}_2 - \mathbf{c}_3), \quad (5.24)$$

where  $\mathbf{c}_1$ ,  $\mathbf{c}_2$ ,  $\mathbf{c}_3$  are randomly selected without replacement from the population  $\mathbf{C}_g$  and  $\gamma$  is a constant factor weighting the differential variation  $\mathbf{c}_2 - \mathbf{c}_3$ . If the confidence

$\pi(\mathbf{c}^h)$  in hypothesis  $\mathbf{c}^h$  is higher than  $\pi(\mathbf{c}_1)$ ,  $\mathbf{c}^h$  replaces  $\mathbf{c}_1$  in generation  $\mathbf{C}_{g+1}$ , otherwise  $\mathbf{c}_1 \in \mathbf{C}_{g+1}$ . After  $I$  iterations ensuring convergence the hypothesis / population member  $\mathbf{c}^*$  with the highest confidence is considered to represent the best solution.

To guarantee a detailed balance of proposal and acceptance with respect to the fitness function  $\pi(\cdot)$  Eq. 5.24 is modified to

$$\mathbf{c}^h = \mathbf{c}_1 + \gamma(\mathbf{c}_2 - \mathbf{c}_3) + \mathbf{k} \quad (5.25)$$

where  $\mathbf{k}$  is drawn from the normal distribution  $\mathbf{k} \sim N(0, \mathbf{a})$  with variance  $\mathbf{a}$  small compared to the variance of population  $\mathbf{C}_g$  [Braak06].

The Differential Evolution update process for a 2D example is visualized in Fig. 5.6.

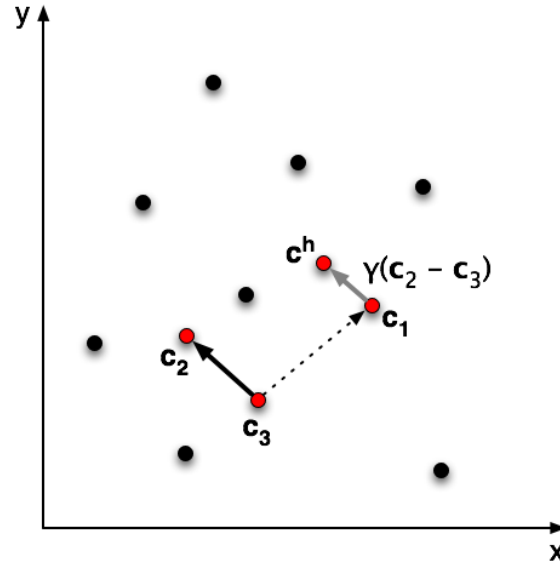


Figure 5.6.: A new hypothesis  $\mathbf{c}^h$  is generated by calculating the vector from  $\mathbf{c}_3$  to  $\mathbf{c}_2$ , weighting it with  $\gamma$  and adding it to  $\mathbf{c}_1$  (Eq. 5.24). The parameter vectors  $\mathbf{c}_1$ ,  $\mathbf{c}_2$ ,  $\mathbf{c}_3$  are randomly selected without replacement from the population  $\mathbf{C}_g$  and  $\gamma$  is a constant factor weighting the differential variation  $(\mathbf{c}_2 - \mathbf{c}_3)$ .

## 5.2. Training

Similar to the Shape Particle Filter algorithm outlined in Chapter 4 the local shape and appearance of the object of interest on a given set of  $N_S$  training images are learned. Again a shape model is constructed from manually annotated shape contours, that form the shape vectors  $\mathbf{v}_i$  ( $i \in 1, \dots, N_S$ ).

Two approaches to automatically generate the previously manually defined region map are introduced, namely automatic region maps and per-pixel region maps. Both are based on the shape model's mean shape  $\bar{\mathbf{v}}$ . In the following the variable  $R_a$  denotes an automatic region map and  $R_p$  denotes a per-pixel region map.

**Automatic Region Map** The manual definition of the region map represents the introduction of a strong bias, as there is no guarantee that their definition is beneficial or at least suited to the convergence and accuracy of the optimization scheme. In contrast to previous approaches a method which derives an optimal region map directly from the training image data is proposed. It takes into account the discriminative power of the computed image features and reflects their distribution in the region map.

Similar to the approach in Chapter 4 an average feature vector  $\mathbf{f}_i$  (Gabor Jets or Monogenic Signal) is computed for each pixel  $i$  within the mean shape (as well as within a border around the mean shape) from the corresponding features  $\mathbf{f}_i^n$  from each training image  $n \in 1 \dots N_S$ . The task is now to estimate a sensible partition, i. e. region map, of the inner and outer/border of the mean shape such that the probabilities  $P_l$  ( $l \in 1, \dots, N_L$ ) of the image classification using this partition convey information, which makes the particle filter converge at a good segmentation. The number of regions within the mean shape is denoted by  $N_{inner}$  and the number of regions within the border by  $N_{border}$ .

First the feature matrix  $\mathbf{F}_n$  for each training image is extracted. Then the mean feature matrix

$$\bar{\mathbf{F}}(j) = \frac{\sum_{n=1}^{N_S} \mathbf{F}_n(W(j, \mathbf{v}_n))}{N_S} \quad (5.26)$$

is calculated from the individual feature maps  $\mathbf{F}_n(W(j, \mathbf{v}_n))$  after each pixel  $j$  of the training images has been warped from shape  $\mathbf{v}_n$  onto the mean shape  $\bar{\mathbf{v}}$ . The area inside the mean shape is clustered into  $N_{inner}$  regions and the corridor is clustered into  $N_{border}$  regions using  $k$ means clustering [MacQueen67]. This algorithm was chosen because it is simple to implement, fast and allows to specify the desired number of clusters. The

advantage of these clustered regions lies in their ability to incorporate similar areas of appearance in the image to one region. Furthermore it is possible that several separated areas are combined to one region. The resulting mean region map  $\bar{R}_a$  with the automatic generated  $N_L = N_{inner} + N_{border}$  regions is warped back to each training shape using TPS warping, so that for each region a distribution of the corresponding descriptors/features can be estimated by sampling. A flowchart of the generation of the automatic region map is shown in Fig. 5.7 and examples for automatic generated region maps for the metacarpal bones and hearts data set can be found in Fig. 5.8.

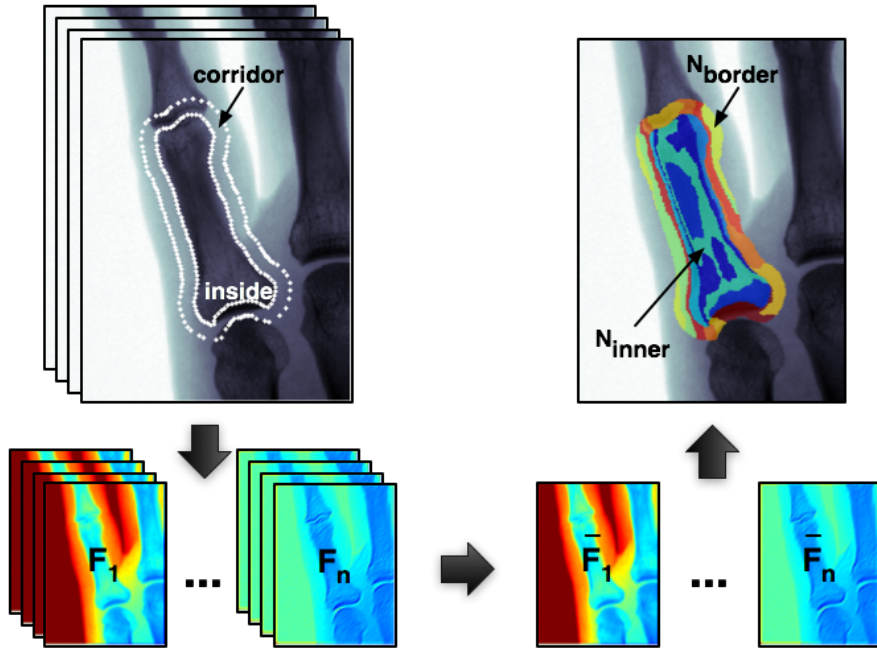


Figure 5.7.: An automatic region map is generated by extracting the feature matrix  $\mathbf{F}_n$  for each of the  $n$  training images, calculating the mean feature matrix  $\bar{\mathbf{F}}_n$  and clustering the area inside the mean shape and the corridor into  $N_L = N_{inner} + N_{border}$  regions.

The location and arrangement of the automatically generated regions results from the incorporation of the feature information of the structure and its surroundings. They capture the necessary level of detail to cope with nearby or overlapping similar structures by placing distinctive regions in crucial image areas. In other words the automatic regions provide a direct representation of the features space, ensuring that similar features are captured with unique regions. The resulting automatic regions are spread throughout the object of interest allowing even inconsistent regions, i. e. same regions at different locations. When using manual region maps a unique region map representation cannot be guaranteed. This means that different regions can represent areas with identical

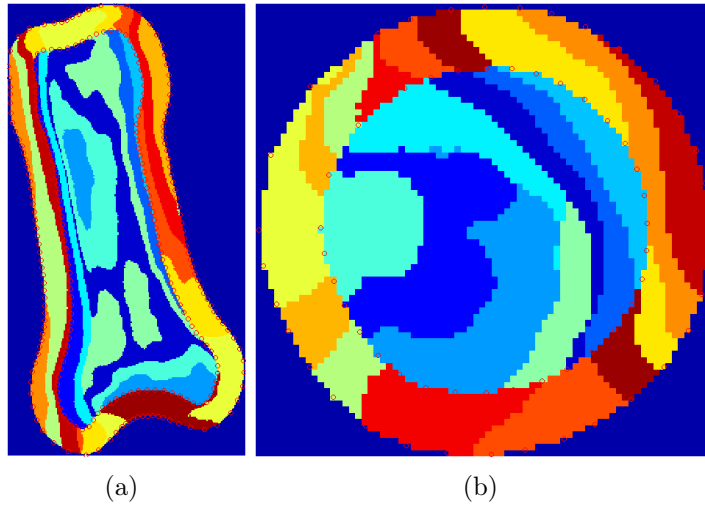


Figure 5.8.: Examples of automatically generated region maps for the metacarpal bones (a) and the hearts data set (b).

features or feature overlaps leading to unreliable classification results.

Details on the evaluation of the automatic determination of region map regions can be found in Sec. 6.2.2. This contribution has been published in [Fischer09].

**Per-Pixel Region Map** Instead of calculating automatic regions inside and around a shape, per-pixel region maps work in a different way. In the strict sense they do not represent an actual region map any more. Based on the mean shape  $\bar{\mathbf{v}}$  of the training set all or a certain percentage of pixels inside of  $\bar{\mathbf{v}}$  are used for hypothesis confidence computation. The confidence is computed pixel-wise, so no region clustering is necessary. In contrast to automatic region maps only pixels within the mean shape contours are used, forming the per-pixel region map  $R_p$ . Furthermore the computational costly image classification step can be omitted.

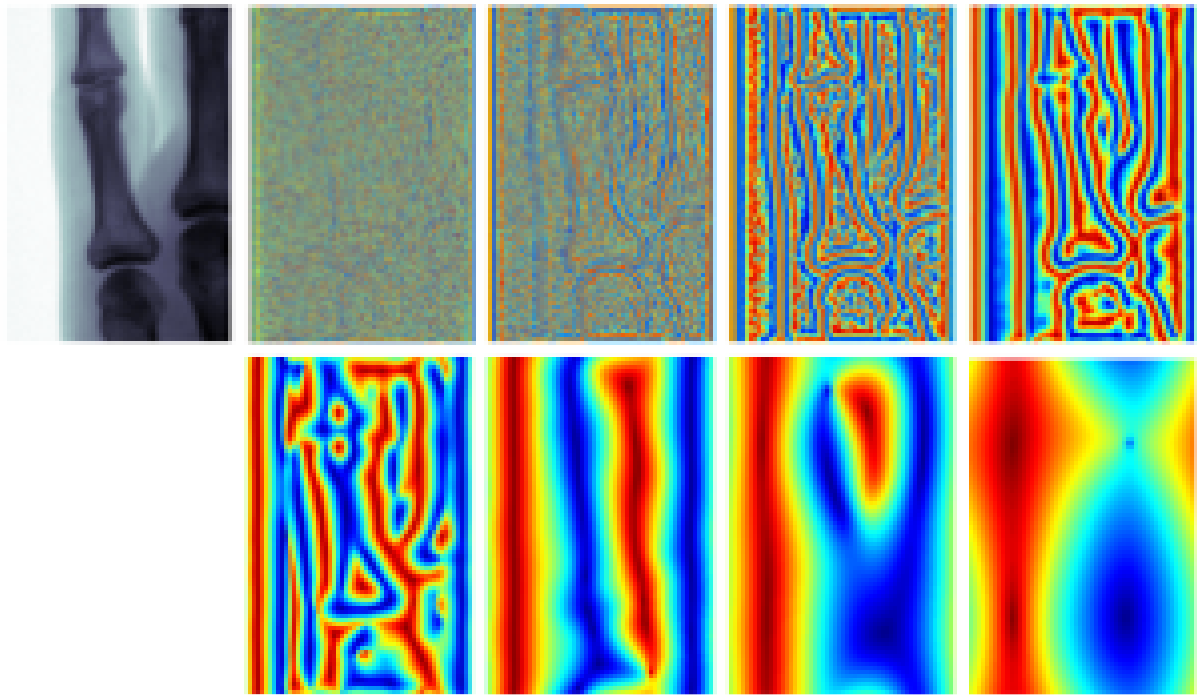
The major advantage of this approach manifests itself later in the shape detection step. The extracted features of the test shape are warped from the mean shape onto newly generated shapes, but only at the pixels considered in the per-pixel region map. Using the result of this image warp, the distance to the mean features at the same pixel coordinates is computed forming the residual used for the Differential Evolution update. The distance can be calculated in different ways. In the scope of this thesis a comparison of

the generated results using euclidean distance, mahalanobis distance and the correlation coefficient is conducted (see Sec. 6.2.1).

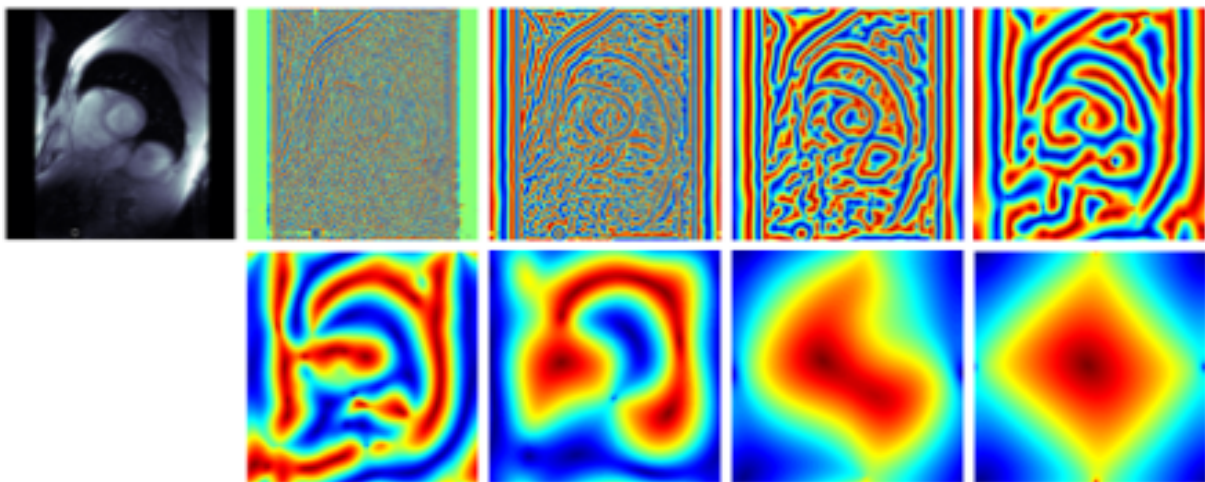
In the following the combination of per-pixel region maps and features extracted using the Monogenic Signal are termed *Improved Monogenic Features* (IMF).

**Image Feature Extraction** In contrast to previous Particle Filter implementations where for example Gabor Jets are used to determine underlying feature information of images, in this thesis the Monogenic Signal is applied. This decision was made due to the promising results and the growing number of publications in this field [Felsberg01, Felsberg02, Pan06, Takaya07, Wietzke08, Kokkinos08, Wietzke09, Rajpoot09]. The following advantages arise: Features extracted based on the Monogenic Signal are invariant to changes in contrast and brightness. Furthermore they contain the phase and the direction information to edges for each pixel, including the actual location of an edge as well as the distances to that edge. The number of features only depends on the selected scale leading to a significant reduction of the number of features compared to Gabor Jet features, whereas the feature number additionally depends on the chosen number of rotation angles and frequencies (see Sec. 4.1.1). A mathematical formulation of the Monogenic Signal can be found in Sec. 5.1.1. Example features using different scales for both the metacarpal bones and hearts data set are shown Fig. 5.9. By taking a closer look on the different filter responses it can be noticed that first, when using small wavelengths, local edges and symmetries are detected. With increasing wavelength the detailed edge information is replaced by larger structures, e.g. the metacarpal bone. With further increase no structural or symmetry information is obtained. Therefore it is crucial to select the right number of wavelengths to achieve a tradeoff between detailed edge information and information concerning whole objects in the image.





(a)



(b)

Figure 5.9.: Example Monogenic features for the metacarpal bones (a) and the hearts (b) data sets. The features were extracted using eight different scales, starting at a wavelength of 4 and successively multiplying it by 2 leading to the following filter wavelengths: 4, 8, 16, 32, 64, 128, 256, 512.

### 5.3. Search

Similar to the original Shape Particle Filter the actual segmentation based on the previous learned features, the region maps and shape model is conducted during the search part. However the search process steps differ according to the region map used. When using automatically generated region maps an image feature classification step yielding pixel probabilities is needed before the Differential Evolution based segmentation algorithm is applied. By relying on a per-pixel region map the classification step can be completely omitted. This can be achieved by directly computing per-pixel similarity measures using Monogenic Signal based features.

**Image Feature Classification** Three methods are implemented to allow the classification of a test image using the information of the extracted training features and the automatic generated region maps yielding probability maps  $P_l$ . As described in Sec. 4.3  $P_l$  contains the probability for each pixel  $j \in 1, \dots, N_P$  in the image to belong to region  $l \in 1, \dots, N_L$ . Due to the amount of features resulting from the feature extraction process the simple  $k$ -NN approach (see Sec. 4.1.2) lead to considerably longer classification time. This can be reasoned by the fact that the  $k$ -NN algorithm calculates the euclidean distance of each test point to all surrounding points in the training set. That means that the larger the training set, the longer the algorithm needs to compute the classification result. Therefore a  $kd$ -Tree (see Sec. 5.1.2) formerly introduced by Bentley in 1975 [Bentley75] was used for faster classification. When classifying the test feature set a  $k$ -Nearest Neighbors approach is employed using the  $kd$ -Tree for a first step in which approximately 100 feature vectors are queried around a test vector. With this subset a standard  $k$ -NN classifier using euclidean distance is employed. Despite a considerably speedup even the  $kd$ -Tree implementation had to be replaced by a faster classification algorithm, namely a linear Support Vector Machine (see Sec. 5.1.3). A comparison of the mean classification time for all three classifiers and three different data sets (synthetic rectangles, metacarpal bones and hearts) can be found in Tab. 5.1. Important to mention is the fact that the better part of the classification runtime is located at different points in the algorithms. When using SVMs the most time consuming part is the training, whereas for the other two algorithms, especially the  $k$ -NN with the  $kd$ -Tree search, the classification step takes longest. In practical application with large sets of images the SVM approach would be the best choice due to the fact that the training is done offline i. e. only once.

	$k$ -NN	$kd$ -Tree	linear SVM	speed gain vs.	
				$k$ -NN	$kd$ -Tree
Synth. rectangles	32.6	20.5	4.2	7.8x	4.9x
Hearts	239.8	134.8	19.9	12.1x	6.8x
Metacarpal bones	641.4	363.8	70.5	9.1x	5.2x

Table 5.1.: Mean classification speed over 10 runs for one test image in seconds for the  $k$ -NN, the  $k$ -NN with a  $kd$ -Tree search and the linear SVM. In the right most column the speed gain achieved by using SVMs compared to the other classification algorithms is shown (SVM times faster than  $k$ -NN and  $k$ -NN with  $kd$ -Tree).

For the actual improved Shape Particle Filtering scheme *no* classification step is needed leading to a major increase in computational performance (see Sec. 5.3).

**Segmentation** In the scope of this thesis the optimization is performed using the DEMCMC (Differential Evolution Markov Chain Monte Carlo) formulation introduced by [Braak06], that uses Differential Evolution [Storn97] for the sampling step in sequential Monte Carlo Methods. The DE algorithm estimates the posterior distribution of the shapes given the image using a similar fitness function as described in Sec. 4.3. Initially, a random set of shapes, the particles, represented by shape parameter vectors  $\mathbf{c}_i$  are generated according to the distribution of the prior shape model.

The region map is deformed according to the shape parameters and for each region the cumulative probability is computed by summing up the probabilities from the corresponding probability map.

By this, a weight can be applied to each particle considering their likelihood, defining the fitness function  $\pi(\mathbf{c}_i)$  (see Sec. 4.3, Eq. 4.8).

Using the per-pixel region map the fitness function is calculated differently. The extracted features of the current test shape hypothesis are warped to the mean shape, but only at the pixel coordinates stored in the per-pixel region map.

The actual fitness function can be calculated using three different methods:

1. **Euclidean distance:** By computing the mean euclidean distance of the resulting test feature vector  $\mathbf{f}_i$  to the mean features at the same pixel coordinates the fitness function is defined as follows:

$$\pi(\mathbf{c}_i) = \frac{\sum_{i=1}^{N_{inside}} (\bar{\mathbf{F}}_i - \mathbf{f}_i)}{N_{inside}}, \quad (5.27)$$

where  $N_{inside}$  is the number of pixels inside the shape defined by the per-pixel region map,  $\bar{\mathbf{F}}_i$  is the mean feature vector of all region map pixels and  $\mathbf{f}_i$  is the warped test feature vector at pixel  $i$ .

2. **Mahalanobis distance:** Again the distance of the resulting test feature vector to the mean features at the region map pixel coordinates are calculated, but this time using the Mahalanobis distance leading to the fitness function:

$$\pi(\mathbf{c}_i) = 1 - e^{\sum_{i=1}^{N_{inside}} (\bar{\mathbf{F}}_i - \mathbf{f}_i) \Sigma^{-1} (\bar{\mathbf{F}}_i - \mathbf{f}_i)}, \quad (5.28)$$

where  $N_{inside}$  is the number of pixels inside the shape defined by the per-pixel region map,  $\bar{\mathbf{F}}_i$  is the mean feature vector of all region map pixel,  $\Sigma$  is  $cov(\bar{\mathbf{F}}_i, \mathbf{f}_i)$  and  $\mathbf{f}_i$  is the warped test feature vector at pixel  $i$ .

3. **Correlation coefficient:** The third way to calculate the fitness function is to compute the correlation coefficient of the resulting test feature vector and the mean features at the region map pixel coordinates:

$$\pi(\mathbf{c}_i) = \sum_{i=1}^{N_{inside}} \frac{1}{N_{inside}} \frac{cov(\mathbf{f}_i, \bar{\mathbf{F}}_i)}{\sqrt{cov(\mathbf{f}_i, \mathbf{f}_i) cov(\bar{\mathbf{F}}_i, \bar{\mathbf{F}}_i)}}, \quad (5.29)$$

where  $N_{inside}$  is the number of pixels inside the shape defined by the per-pixel region map,  $\bar{\mathbf{F}}_i$  is the mean feature vector of all region map pixel and  $\mathbf{f}_i$  is the warped test feature vector at pixel  $i$ .

Test runs described in Sec. 6.2.1 where conducted to select the fitness function leading to the most accurate segmentation results.

New particles are finally generated from the current set of particles using the Differential Evolution algorithm detailed in Sec. 5.1.4, e. g. the update rule in Eq. 5.24 visualized in Fig. 5.6.

# Chapter 6.

## Experiments

In this chapter all experimental results are presented. First Sec.6.1.1 describes the experimental setup including the data sets used for evaluation and Sec.6.1.2 presents the testing environment. Sec.6.2 gives a detailed presentation of the achieved results using the approaches described above.

Because of the small amount of training images and therefore training features, LOOCV was used for evaluation. In each test run one object of interest (e.g. synthetic rectangle, metacarpal bone, heart and lung) was used for testing and the remaining objects were used for training. This was repeated till each object was once used as test data. The following percentages of the region map were used: 1%,5%,10%,20%,...,100%. The LOOCV runs were repeated 12 times for each input image type, but using only a certain percentage of the respective region map.

Five different segmentation setups were used:

1. **IMF:** The improved Shape Particle Filtering scheme using a per-pixel region map based on Monogenic features (IMF = Improved Monogenic Features).
2. **GaborIn:** An automatic region map using only regions inside the shape is generated by clustering Gabor Jet feature responses.
3. **GaborIO:** An automatic region map is generated using regions inside the shape and in a corridor around the shape. Based on Gabor jets.
4. **MonoIn:** An automatic region map using only regions inside the shape is generated based on Monogenic features.

5. **MonoIO:** Again an automatic region map is generated using regions inside the shape and in a certain corridor around the shape based on Monogenic features.

From each resulting shape of the DE the euclidean distance, more precisely the landmark error to the hand annotated ground truth is calculated and visualized using box-and-whisker plots (see Sec. 6.2). The landmark error computation is done in two ways:

- **Landmark to Landmark Error:** This is the euclidean distance between a point or landmark of the resulting shape and its equivalent in the annotated ground truth.
- **Landmark to Contour Error:** Depicts the shortest distance between a point or landmark of the resulting shape to the contour outlining the annotated ground truth landmarks.

The two landmark errors are visualized in Fig. 6.1.

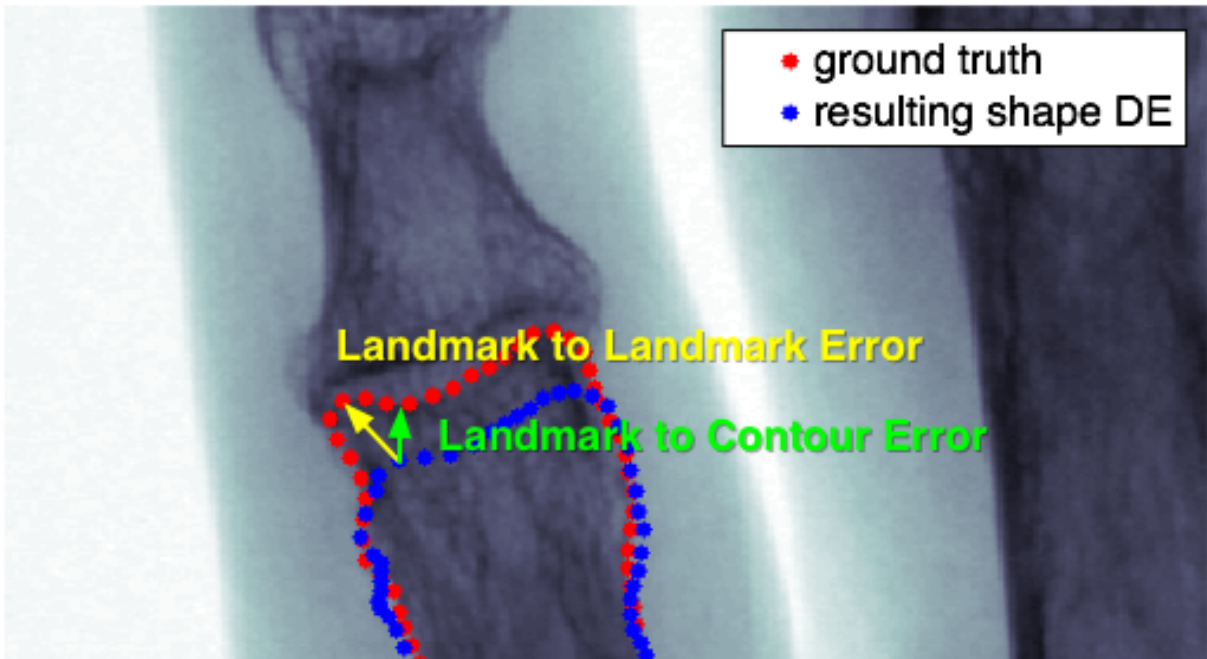


Figure 6.1.: Example segmentation result of the metacarpal bones data set with landmark to landmark error (yellow) and landmark to contour error (green) between the manually annotated ground truth (red) and the resulting shape (blue) of the segmentation process.

## 6.1. Setup

Evaluation was performed with a MATLAB implementation on synthetic 2D images, on metacarpal bone radiographs, on MRI slices of the heart and on CT slices of the lung. All approaches used a Differential Evolution based Shape Particle filtering strategy in contrast to the originally proposed importance resampling.

### 6.1.1. Data Sets

- **Synthetic Images** Seven synthetic images with  $100 \times 100$  pixels each containing a rectangle were created. The rectangles change size along the  $x$ -axis, constituting the first and only mode of shape variation. Example images are shown in Fig. 6.2(a).
- **Metacarpal Bone Radiographs** The second evaluation data set was a collection of 15 radiographs of human metacarpal bones with the resolution of approximately  $500 \times 400$  pixels each. An example image is shown in Fig. 6.2(b).
- **MRI Slices of the Heart** 14 short-axis, end-diastolic cardiac MRI slices of the human heart with the resolution of  $256 \times 256$  pixels with manually placed landmarks on the epicardial and endocardial contours [Stegmann01]. An example image is shown in Fig. 6.2(c).
- **CT Slices of the Lung** 13 CT slices of the lung with the left lung and 15 CT slices of the lung with the right lung annotated. The images of both data sets had a resolution of  $512 \times 512$  pixels each. Left and right lungs were used as two distinct data sets. An example image is shown in Fig. 6.2(d).

Example images for all data sets including the manually annotated ground truth are shown in Fig. 6.2.

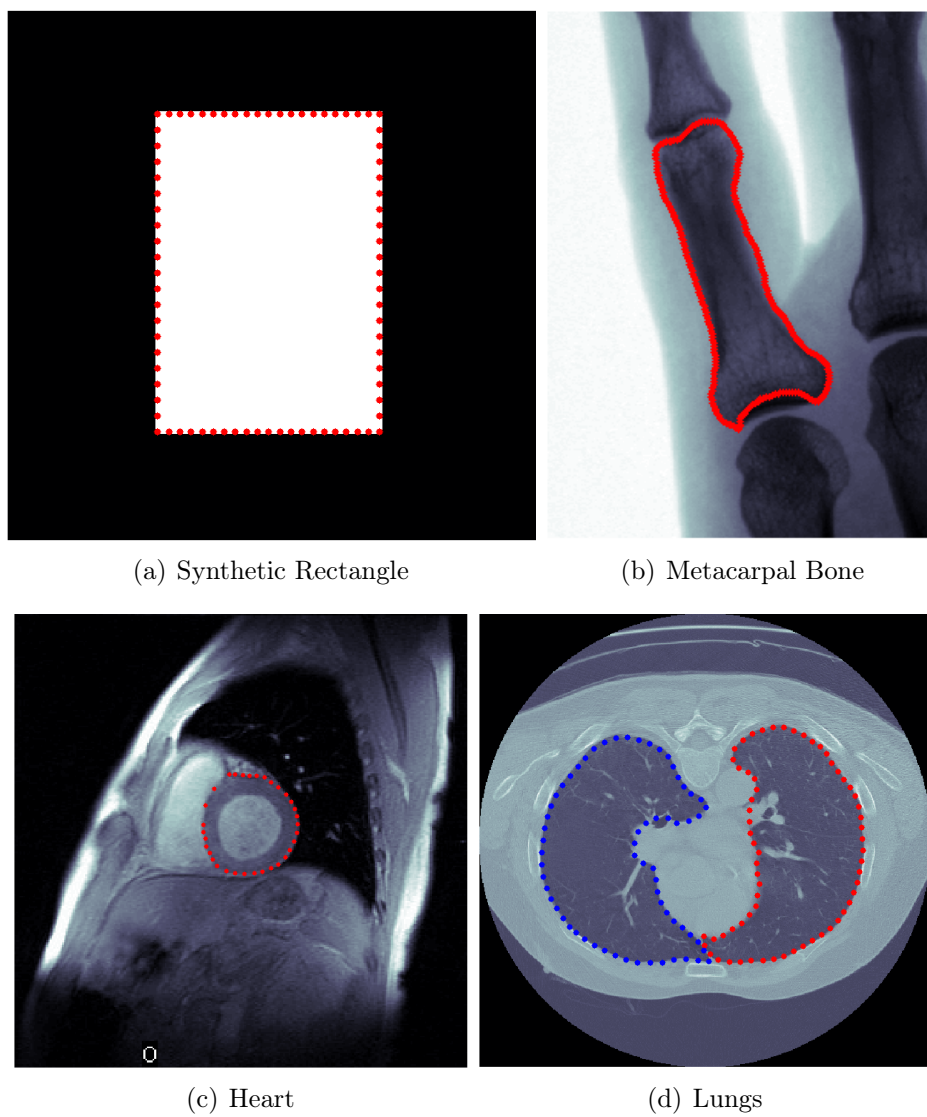


Figure 6.2.: Example images of all 2D data sets with manually annotated ground truth.



## 6.1.2. Testing Environment

In this section the experimental setup is described including the employed hard- and software, as well as the parameters used during the segmentation scheme.

**Parameter Setting** In all runs with all five different segmentation approaches, 200 Differential Evolution iterations were performed. Preliminary experiments showed, that the algorithm always converged when using 200 iterations. The parameters used in the different steps of the segmentation pipeline for all data sets were obtained by applying LOOCV and are shown in Tab.6.1. For the left and right lungs data set separate runs were conducted using the same parameter setup. The parameters are defined as follows:

- $\gamma$  = Differential Evolution weight (see Sec. 5.1.4, Eq. 5.24)
- $in$  = number of inner regions  $N$  inner and  $out$  = number of outer regions  $N$  border for the automatic generation of the region map (see Sec. 5.2), as well as the size of the region around the shape denoted in the table as "R. size".
- $s$  = scale/filter size,  $f$  = filter frequency and  $\theta$  = filter angle for the Gabor Jet feature extraction scheme (see Sec. 4.1.1)
- $\#s$  = number of scales,  $\lambda$  = minimal wavelength used,  $\mu$  = multiplication factor of successively filters and  $\sigma$  = ratio of the standard deviation of the Gaussian describing the log Gabor filter's transfer function in the frequency domain to the filter center frequency for the Monogenic Signal based feature extraction scheme (see Sec. 5.1.1)

**Hard- and Software** Hardware details of the computers used to generated the results presented below are shown in Tab.6.2. The table includes all technical information needed to reproduce the computer testing environment, namely the processor type and clock rate in GHz, the built in main memory type and size, as well as the installed operating system. All results were generated using the MacPro, the DELL XPS720 and the DELL PowerEdge 2900 running MATLAB R2009b.

Chapter 6. Experiments

		DE	Regions		R.	Gabor Jets			Monogenic			
		$\gamma$	<i>in</i>	<i>out</i>	Size	<i>s</i>	<i>f</i>	$\theta$	# <i>s</i>	$\lambda$	$\mu$	$\sigma$
Rectangles	GaborIn	0.85	7	0	0	10,20,40	1,2,4	0,45,90,135	5	4	2	0.65
	GaborIO		7	7	10							
	MonoIn		7	0	0							
	MonoIO		7	7	10							
	IMF		0	0	0							
Handbones	GaborIn	0.85	7	0	0	10,20,40	1,2,4	0,45,90,135	6	4	2	0.65
	GaborIO		7	7	20							
	MonoIn		7	0	0							
	MonoIO		7	7	20							
	IMF		0	0	0							
Hearts	GaborIn	0.85	7	0	0	10,20,40	1,2,4	0,45,90,135	6	4	2	0.65
	GaborIO		7	7	10							
	MonoIn		7	0	0							
	MonoIO		7	7	10							
	IMF		0	0	0							
Lungs	GaborIn	0.85	7	0	0	10,20,40	1,2,4	0,45,90,135	6	4	2	0.65
	GaborIO		7	7	10							
	MonoIn		7	0	0							
	MonoIO		7	7	10							
	IMF		0	0	0							

Table 6.1.: In this table the parameter setting for all 2D data set runs are presented. The same parameters were used for the separate left and right lung segmentation runs. Please refer to the text above for details on the parameters.

	Processor			Memory		OS
	Type	GHz	Type	GB		
MacPro	Quad-Core Intel(R) Xeon(R)	2.26	DDR3	16	OS X 10.6.2	
DELL PowerEdge 2900	Quad-Core Intel(R) Xeon(R)	3.00	DDR3	47.3	Ubuntu 9.10	
DELL XPS720	Intel(R) Core(TM)2 Quad	2.66	DDR3	7.8	Ubuntu 8.10	
MacBook Pro	Intel(R) Core(TM)2 Duo	2.53	DDR3	4	OS X 10.5.8	

Table 6.2.: In this table the computers involved in the result generation are described, including details on the built-in processors, memory and operating system used.

## 6.2. Results

In this section the final results of the proposed segmentation algorithms are presented. Preliminary results concerning the selection of the residual or fitness function used in the improved Shape Particle Filtering scheme, more precisely in the Differential Evolution algorithm are described in Sec. 6.2.1. Tests to determine the optimal number of automatic regions are described in Sec. 6.2.2. Sec. 6.2.3 to Sec. 6.2.7 list the generated segmentation results for all data sets. Results for all data sets considering *landmark to landmark* and *landmark to contour* error are presented as follows:

- Exemplarily results for best and typical segmentation accuracy.
- Comparison boxplot of all five proposed segmentation approaches.
- Comparison plot of the mean landmark error of all five segmentation approaches.
- Boxplots for each approach showing the segmentation result using different percentages of the respective region map.
- Bar plots of the achieved runtime using different percentages of the respective region map.

Boxplots were generated using the MATLAB boxplot implementation. The central mark on each box is the median, the box's edges represent the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually (red +).

The resulting mean *landmark to landmark* and the mean *landmark to contour* error in pixels for all segmentation approaches can be found in Tab. 6.3 for all data sets. It has to be noted that the mean landmark error is strongly influenced by outliers and therefore the values in the table should be considered in combination with the respective result boxplots.

**Landmark Error** Generally it can be observed that approaches using the Monogenic Signal for feature extraction yield better segmentation results (smaller landmark errors). Exceptions are the synthetic rectangles and the right lungs data set. For the synthetic data set this can be reasoned by the simplicity of the shape and the underlying image and therefore the resulting landmark errors for all approaches on this data set are similar. By examining the landmark errors achieved on the right lungs data set it can be noticed

that the GaborIn approach led to better results only at three runs (using 50%, 70% and 90% of the region map). For all other runs the Monogenic Signal based approaches yielded better or at least equal landmark errors.

Furthermore the IMF approach led to better or at least equal landmark errors compared to all other approaches on nearly all data sets. On the hearts data set the MonoInOut approach led to slightly smaller landmark errors (maximum error difference was 0.7 pixel). In the first two runs (1% and 5% of the region map) of the right lungs data set the IMF approach also yielded larger landmark errors than both approaches based on Monogenic features and automatic region maps.

Additionally it can be noted that the approaches relying on automatically generated region maps and using inner and border regions (GaborInOut, MonoInOut) in most cases resulted in smaller landmark errors than those using only inner regions (GaborIn, MonoIn). This can be reasoned due to the inclusion of appearance information in the surrounding of the shape. Using only inner regions therefore lacks the important structural details encoded in the surrounding of the object of interest. Exceptions are the lungs data sets, where both approaches led to similar results. By examining the lungs images it can be noticed that the tissue surrounding the lungs and therefore the respective appearance features are quite homogenous. This lack of additional neighboring information can be the reason that on these data sets both approaches yielded similar segmentation results.

Overall can be noted that features based on the Monogenic Signal yield a better and more distinct description of the structure and appearance in medical images similar to the data sets used. Approaches using inner and border regions in the region map are best applied on images, where the object of interest is surrounded by a multitude of different structures. On images with a homogenous surrounding of the object, region maps with only inner regions are sufficient.

Furthermore can be noticed that the segmentation of objects with complex anatomy (e. g. metacarpal bones and especially lungs) yielded larger landmark errors compared to simple structured objects (e. g. hearts) due to the applied too rigid shape model. The shape model therefore constrains the segmentation result on complex structures.

**Runtime** The mean runtime in minutes needed to segment one object during a LOOCV run using a certain percentage of the respective region map for all five segmentation approaches and all data sets are presented in Tab. 6.4. The runtime is furthermore visualized in Fig. 6.3. It has to be noted that the segmentation runs for the different data sets were conducted on different computers (see Sec. 6.1.2). That means the significant information when comparing the runtime of different segmentation runs is the times an approach was faster than another and not the actual runtime values. Due to this fact a runtime comparison between different data sets was not undertaken.

Generally it can be observed that the runtime increases proportionally to the percentage of the respective region map, regardless of the selected segmentation approach and data set (see Fig. 6.3). That means the more pixel are used of the respective region map, the longer the algorithm needs to converge. This shows that the entirety of pixels of the region map is an important factor concerning the overall computational performance. The number of pixels affects the image warp conducted during the Differential Evolution process, i. e. the more pixel are warped, the slower the algorithm gets. Furthermore the runtime of the algorithm's training phase is slowed down with a growing number of region pixels used. This results on the one hand from the region clustering when using automatic region maps and on the other hand from the classification step when using SVMs.

By taking a closer look on the results it can be observed that the IMF approach outperformed all other approaches on the synthetic rectangles and the hearts data sets at all runs. On the remaining data sets it yielded faster segmentation results than all approaches based on region maps with inner and border regions (GaborInOut, MonoInOut). Compared to the approaches using only inner regions (GaborIn, MonoIn) the IMF approach was slower after the 20% run on the metacarpal bones data set, after the 40% run on the left lungs data set and after the 10% run on the right lungs data set.

Overall can be said that approaches using inner and border regions (GaborInOut, MonoInOut) yielded the longest runtimes. In these approaches longer training or test phases are needed for the respective classification algorithm i. e. SVMs due to the large number of regions in the region map. The advantage of the IMF approach is that it does not need the time-consuming classification step, that is necessary in all approaches using automatic region maps. Comparing approaches using both inner and border regions with those using only inner regions the different runtime results can be explained

## Chapter 6. Experiments

by the different number of regions used. A large number of regions again affects the classification runtime, when using SVMs, as well as yields slower image warps during the DE. Furthermore approaches using features based on the Monogenic Signal have the advantage that fewer features compared to Gabor Jets are used. That in turn influences the classification speed. The more features are used, the larger are the dimensions of the training and test sets and therefore the distance calculations are more costly yielding longer classification times.

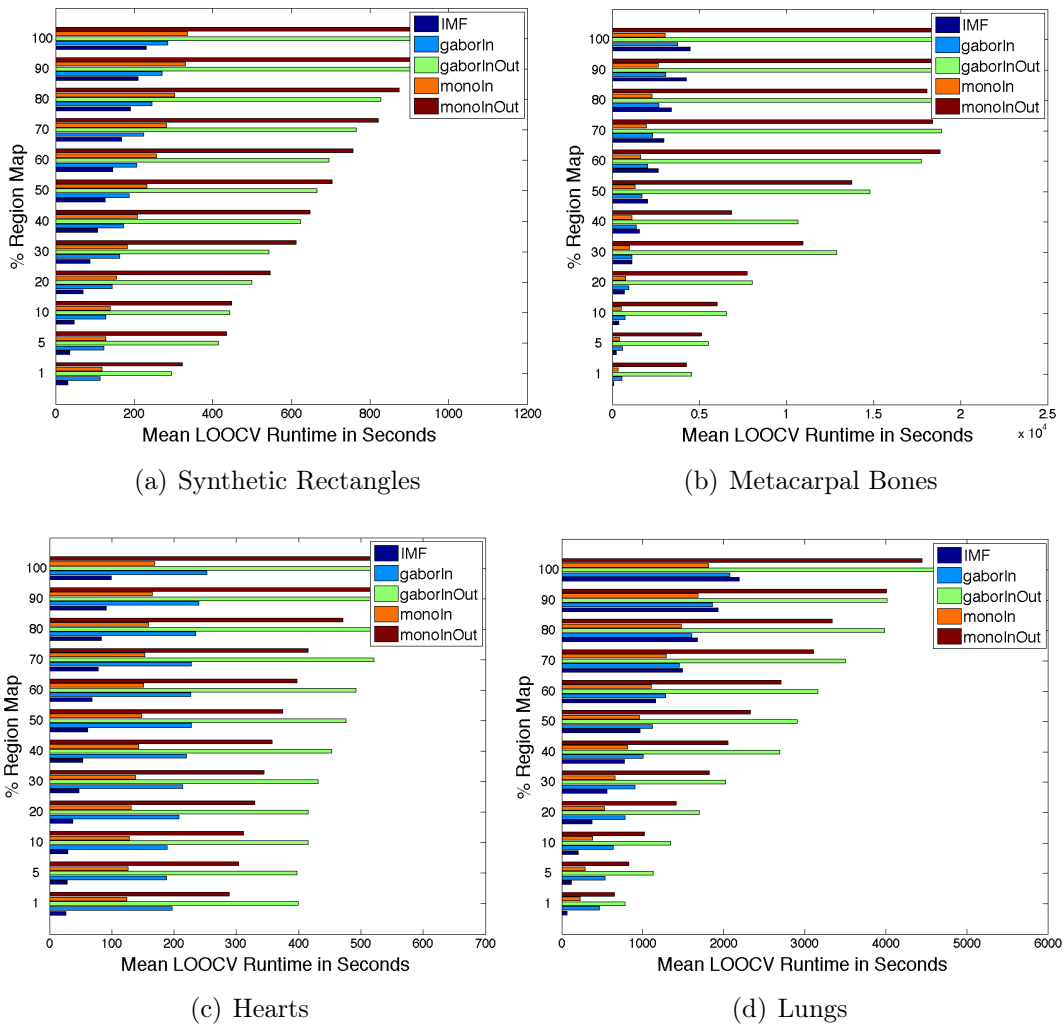


Figure 6.3.: Bar plots of the mean runtime in seconds needed to segment an image using different percentages of the respective region map and for all applied segmentation methods for all four data sets.

		% region map												
1		5	10	20	30	40	50	60	70	80	90	100		
$\bar{x}$	$\bar{y}$	$\bar{x}$	$\bar{y}$	$\bar{x}$	$\bar{y}$	$\bar{x}$	$\bar{y}$	$\bar{x}$	$\bar{y}$	$\bar{x}$	$\bar{y}$	$\bar{x}$	$\bar{y}$	
GaborIn	0.9	1.6	0.6	1.6	0.6	1.6	0.7	1.5	0.7	1.5	0.6	1.6	0.7	1.6
GaborIO	1.5	1.9	<b>0.3</b>	1.5	0.5	1.6	<b>0.3</b>	<b>1.5</b>	0.4	<b>1.5</b>	<b>0.4</b>	<b>1.4</b>	0.4	<b>1.5</b>
MonolIn	2.1	1.9	0.4	1.5	0.5	1.6	0.5	<b>1.5</b>	0.4	<b>1.5</b>	0.5	1.5	0.4	<b>1.5</b>
MonolIO	1.5	1.9	1.8	2.0	1.7	1.9	<b>0.3</b>	<b>1.5</b>	1.4	1.8	<b>0.3</b>	<b>1.5</b>	0.4	<b>1.4</b>
IMF	<b>0.7</b>	<b>1.4</b>	0.5	<b>1.5</b>	0.5	1.5	0.5	<b>1.5</b>	0.5	<b>1.5</b>	0.4	1.6	0.5	<b>1.5</b>
GaborIn	12.1	3.7	13.3	3.9	9.7	3.6	11.4	3.6	11.5	3.5	13.4	3.8	11.5	3.6
GaborIO	9.9	3.9	9.0	3.6	8.2	3.6	9.1	3.6	8.5	3.5	8.4	3.5	8.3	3.5
MonolIn	9.9	4.1	8.8	3.7	7.3	3.0	7.0	2.9	7.4	3.1	8.1	3.3	7.5	3.1
MonolIO	9.2	3.3	7.9	<b>2.6</b>	8.3	<b>2.7</b>	8.2	2.8	8.4	<b>2.6</b>	7.8	<b>2.5</b>	8.4	2.7
IMF	<b>7.9</b>	<b>2.9</b>	<b>7.5</b>	2.8	<b>6.7</b>	<b>2.7</b>	<b>6.9</b>	<b>2.6</b>	<b>6.8</b>	<b>2.6</b>	<b>6.7</b>	2.7	<b>6.8</b>	<b>2.6</b>
GaborIn	39.4	28.8	34.3	25.6	24.1	18.4	32.8	25.4	32.9	25.7	33.3	26.0	33.1	25.8
GaborIO	22.6	16.7	18.7	13.2	21.2	15.8	20.8	15.5	20.9	15.4	20.9	15.5	20.8	15.3
MonolIn	5.3	2.1	4.8	1.8	5.1	1.7	4.5	1.6	4.6	1.7	4.7	1.6	4.6	1.6
MonolIO	<b>4.2</b>	<b>1.8</b>	<b>3.6</b>	<b>1.6</b>	<b>4.4</b>	<b>1.5</b>	<b>3.9</b>	<b>1.5</b>	<b>3.7</b>	<b>1.4</b>	<b>3.8</b>	<b>1.5</b>	<b>3.7</b>	<b>1.5</b>
IMF	4.3	<b>1.8</b>	4.3	1.7	<b>4.4</b>	1.7	4.2	1.7	4.2	1.7	4.2	1.7	4.2	1.7
GaborIn	20.4	6.1	20.4	5.7	19.4	5.4	18.5	5.6	18.3	4.7	15.7	4.6	17.3	4.7
GaborIO	18.2	5.4	16.5	5.3	17.9	4.7	18.2	5.0	17.0	5.5	18.9	5.1	22.5	5.3
MonolIn	18.1	5.2	16.9	4.7	18.6	5.3	<b>17.2</b>	4.8	17.9	5.1	17.2	4.9	18.9	4.8
MonolIO	<b>17.6</b>	<b>5.0</b>	18.6	4.8	19.1	5.1	18.1	4.7	15.3	5.3	16.4	4.7	17.9	4.4
IMF	17.7	<b>5.0</b>	<b>15.6</b>	<b>4.2</b>	<b>15.8</b>	<b>4.1</b>	17.9	<b>4.0</b>	<b>15.0</b>	<b>4.1</b>	<b>14.6</b>	<b>4.1</b>	<b>16.9</b>	<b>4.1</b>
GaborIn	22.0	6.2	18.2	5.3	22.4	4.8	18.6	5.3	17.8	5.2	19.0	<b>4.7</b>	18.2	<b>4.7</b>
GaborIO	24.1	7.5	18.9	6.4	23.5	6.2	21.6	6.3	23.5	7.4	22.5	6.6	20.1	5.9
MonolIn	<b>17.1</b>	5.0	<b>16.5</b>	<b>4.9</b>	17.7	<b>4.6</b>	<b>17.8</b>	<b>4.6</b>	20.2	5.4	19.4	5.3	<b>17.4</b>	4.9
MonolIO	19.5	<b>4.5</b>	16.6	5.6	18.6	5.5	<b>17.8</b>	5.5	18.4	5.4	20.9	5.5	18.5	4.6
IMF	20.0	5.3	17.8	5.2	<b>17.4</b>	5.1	17.9	5.0	<b>16.5</b>	<b>4.8</b>	<b>16.7</b>	<b>4.7</b>	17.8	5.3

Table 6.3.: Mean landmark to landmark ( $\bar{x}$ ) as well as the mean landmark to contour ( $\bar{y}$ ) error for all segmentation approaches and all data sets. The smallest landmark to landmark as well as the smallest landmark to contour error for the different data sets using the five proposed approaches are marked bold.

		% region map											
		1	5	10	20	30	40	50	60	70	80	90	100
Rectangles	GaborIn	1.88	2.04	2.13	2.41	2.72	2.87	3.13	3.44	3.74	4.08	4.51	4.76
	GaborIO	4.92	6.92	7.39	8.32	9.05	10.38	11.08	11.60	12.74	13.80	15.07	16.78
	MonoIn	1.96	2.13	2.31	2.60	3.05	3.48	3.87	4.28	4.70	5.06	5.52	5.58
	MonoIO	5.38	7.26	7.47	9.10	10.20	10.79	11.72	12.63	13.68	14.58	16.3	17.00
	IMF	<b>0.52</b>	<b>0.60</b>	<b>0.80</b>	<b>1.16</b>	<b>1.47</b>	<b>1.78</b>	<b>2.11</b>	<b>2.44</b>	<b>2.79</b>	<b>3.19</b>	<b>3.51</b>	<b>3.85</b>
Hand Bones	GaborIn	9.03	9.9	12.23	15.63	18.97	22.79	28.68	33.69	38.43	44.54	51.39	62.53
	GaborIO	75.60	92.08	109.21	134.24	215.04	177.73	246.74	296.13	315.21	314.54	364.77	393.88
	MonoIn	5.88	6.71	8.57	12.62	<b>16.22</b>	<b>19.12</b>	<b>21.60</b>	<b>27.40</b>	<b>32.79</b>	<b>38.00</b>	<b>44.26</b>	<b>50.66</b>
	MonoIO	70.89	85.48	100.47	129.07	182.35	114.35	229.55	313.69	306.88	301.29	323.53	341.31
	IMF	<b>1.72</b>	<b>3.92</b>	<b>6.09</b>	<b>11.68</b>	19.06	26.31	33.71	43.91	49.41	56.84	71.24	74.89
Hearts	GaborIn	3.28	3.14	3.15	3.47	3.56	3.67	3.79	3.78	3.80	3.91	4.00	4.22
	GaborIO	6.66	6.63	6.93	6.92	7.19	7.55	7.93	8.21	8.68	9.15	8.95	10.32
	MonoIn	2.06	2.10	2.13	2.18	2.30	2.38	2.47	2.51	2.56	2.65	2.75	2.82
	MonoIO	4.82	5.06	5.19	5.50	5.75	5.95	6.24	6.61	6.93	7.86	9.64	9.95
	IMF	<b>0.44</b>	<b>0.47</b>	<b>0.50</b>	<b>0.88</b>	<b>0.79</b>	<b>1.78</b>	<b>1.03</b>	<b>1.14</b>	<b>1.30</b>	<b>1.39</b>	<b>1.53</b>	<b>1.65</b>
Lungs left	GaborIn	7.79	8.91	10.63	13.06	15.11	16.80	18.67	21.31	24.21	26.71	30.96	34.60
	GaborIO	12.99	18.81	22.44	28.36	33.66	44.82	48.46	52.75	58.37	66.44	66.97	83.8
	MonoIn	3.80	4.79	6.31	8.76	10.97	13.56	<b>15.92</b>	<b>18.37</b>	<b>21.49</b>	<b>24.57</b>	<b>28.04</b>	<b>30.27</b>
	MonoIO	10.83	13.85	16.99	23.54	30.34	34.22	38.89	45.17	51.74	55.58	66.87	74.18
	IMF	<b>1.04</b>	<b>2.00</b>	<b>3.37</b>	<b>6.28</b>	<b>9.28</b>	<b>12.88</b>	16.08	19.32	24.81	27.88	32.15	36.47
Lungs right	GaborIn	11.31	13.07	16.35	18.73	19.74	22.15	25.39	28.23	29.40	30.34	36.16	38.00
	GaborIO	16.27	21.99	25.86	32.25	37.85	45.44	49.64	58.82	59.54	77.67	84.62	91.89
	MonoIn	5.61	7.83	7.06	<b>10.36</b>	<b>14.59</b>	<b>16.44</b>	<b>19.24</b>	<b>22.74</b>	<b>25.58</b>	<b>27.58</b>	<b>32.18</b>	<b>36.33</b>
	MonoIO	15.65	17.93	28.72	29.99	31.63	37.87	42.43	47.22	53.01	62.27	71.93	96.62
	IMF	<b>1.40</b>	<b>3.63</b>	<b>5.14</b>	10.84	14.06	17.86	20.80	23.36	28.62	31.94	34.30	39.56

Table 6.4.: The mean runtime in minutes needed to segment one object during a LOOCV run using a certain percentage of the respective region map for all five segmentation approaches and for all data sets. The runtime of the fastest approach is marked bold for each evaluated data set.



### 6.2.1. Residual Selection

In order to ensure the most accurate segmentation results using the proposed improved Shape Particle Filtering scheme, tests were run to obtain the confidence measure best correlated with the landmark error. See Sec. 5.3 for detailed information on the different residual measures. LOOCV runs were performed for the metacarpal bones and the hearts data set. In each run all three measures were computed and plotted against the resulting landmark error. Furthermore the correlation coefficient for each approach is calculated. The resulting values are shown in Tab. 6.5. In Fig. 6.4 the residual comparison plots for the metacarpal bones and the hearts are shown.

Analyzing these results shows that all three methods led to a similar correlation of residual and landmark error. Due to the fact that using the mahalanobis distance led to slightly better correlation, this method was selected to be used in the later evaluation runs on all data sets.

	euclidean distance	mahalanobis distance	correlation coefficient
Metacarpal Bones	0.96856	<b>0.96981</b>	0.96591
Hearts	0.95190	<b>0.95492</b>	0.92644

Table 6.5.: The correlation coefficients computed of the three different residual calculation methods and the resulting landmark error. Analyzing these results shows that all three methods led to a similar correlation of residual and landmark error.

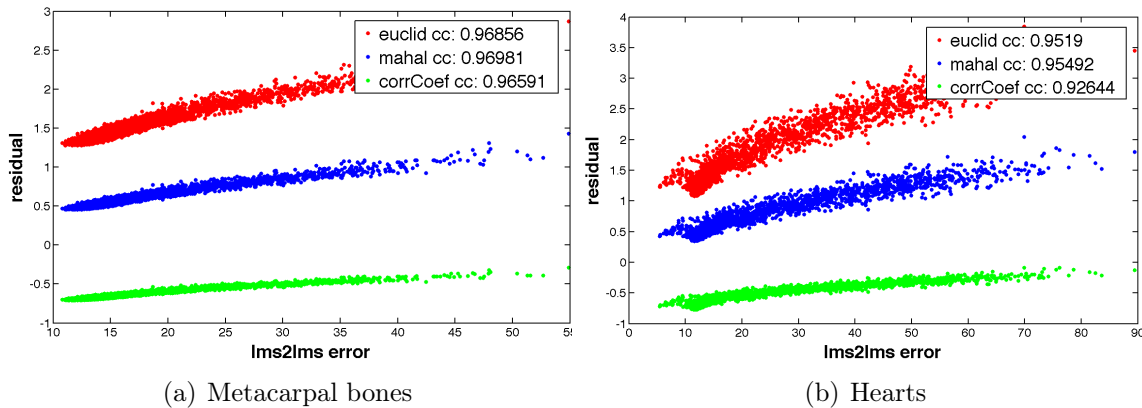


Figure 6.4.: Residual comparison plots for the metacarpal bones (a) and the hearts (b) data sets. The residual calculated (euclidean distance - red, mahalanobis distance - blue, correlation coefficient - green) is plotted against the landmark error.

### 6.2.2. Automatic Region Determination

The tests described below were conducted before the runs, where the proposed optimization steps are evaluated to ensure that segmentation improvements compared to a Shape Particle Filter approach using a manual region map are possible.

The optimal number of regions, i. e. the one resulting in the best segmentation results, is obtained by three LOOCV runs on the metacarpal bones and the hearts data set. For each test shape all 100 possible region number combinations ( $N_{inner} \times N_{border}$  where  $N_{inner}, N_{border} \in 1, \dots, 10$ ) are generated and the segmentation results are computed. By comparing the segmentation results for each region pair and selecting those with the minimal landmark error to ground truth, the optimal number of regions for the region map is determined. Examples of these automatic region maps and for comparison their manual region map counterparts are shown in Fig. 6.5. Visualizations of the mean landmark error of all inner and outer region number combinations for the metacarpal bones and the hearts data set are shown in Fig. 6.6. For the metacarpal bones a auto region map with 5 inner and 7 border regions and for the hearts a region map with 9 inner and 9 border regions led to the best segmentation results and therefore to the minimal landmark error.

The segmentation results are then compared to results achieved using manual region maps. In Fig. 6.7 boxplots of the resulting landmark errors for all four distinctive runs (manual vs. automatic region map each for the metacarpal bones and the hearts) are shown. The approach using automatically generated region maps outperformed the manually defined version especially on images with low input image quality. Due to the better incorporation of the feature information of the structure and its surroundings, the auto region maps lead to more precise classifications and therefore to more exact segmentation results. Furthermore the manual region maps could not provide the necessary level of detail to cope with nearby or overlapping similar structures because of the lack of distinctive regions in crucial image areas. In particular this was observed on the metacarpal bones data set, where the main problem arose at the distal bone region i.e. the joint area where extreme narrow inter-bone spaces appeared.

It can be observed that for both data sets good segmentation results were achieved with region combinations from 5 to 9 inner and border regions. The images show that when using only a small number of regions the underlying feature information cannot be captured yielding larger landmark errors. It is also not sufficient to use a large number of

inner and only a few outer clusters, showing that it is important to include information located in the surrounding of the shape. Furthermore it can be observed that after a certain point (on the evaluated data sets after 9 inner and outer regions) with a growing number of regions the segmentation accuracy decreases. This can be reasoned with a loss of information due to over clustering i. e. splitting regions that contain conjoining information.

In the later experiments each shape is tested only once per LOOCV run and the number of regions in the region map was selected based on the previously presented results, i. e. within 5 to 9 regions. The different number of runs as well as the different parameter selection explains the changes of the achieved mean landmark errors.

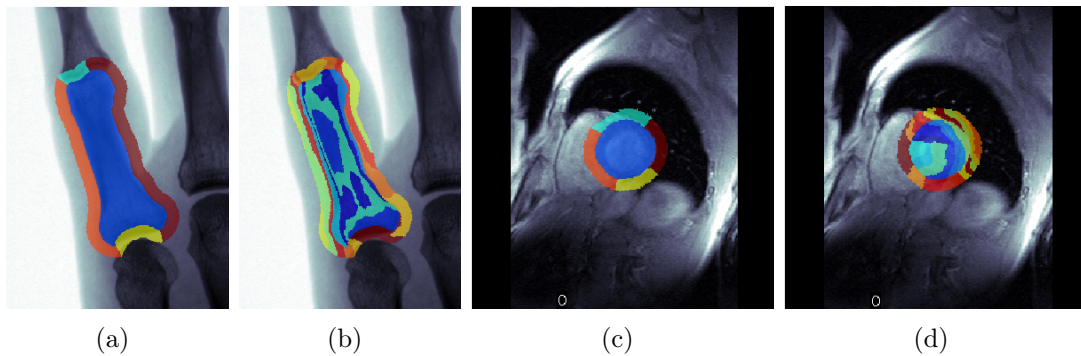


Figure 6.5.: Manually predefined and automatically generated region maps for the metacarpal bones (a),(b) and the hearts (c),(d). The automatic results show how the features (Gabor Jets or Monogenic Signal based) are clustered into regions which are considerably different from the ones chosen by humans. The theoretical background as well as an interpretation of the automatic region maps can be found in Sec. 5.2.

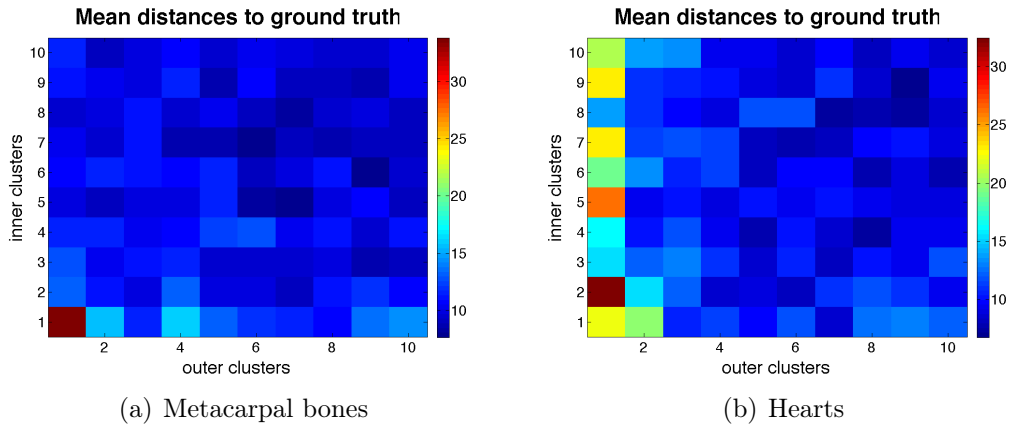


Figure 6.6.: Visualization of the mean landmark error for all evaluated combinations of region numbers over three LOOCV runs for the metacarpal bones (a) and the hearts (b). As can be expected, using a too small number of regions results in poor results as the resulting region probability estimates do not convey enough geometric information for the optimization [Fischer09].

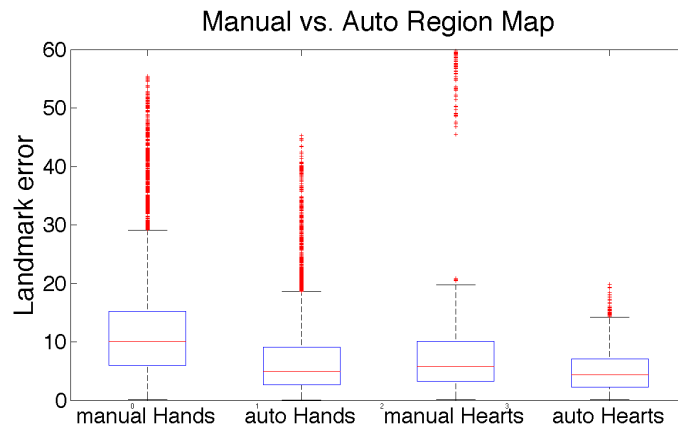


Figure 6.7.: Boxplots of the results for the image segmentation using manually and automatically generated region maps for the metacarpal bones and the hearts data sets. Using the automatic region estimation results in fewer outliers and higher accuracy, displaying the effectiveness of the proposed approach [Fischer09].

### 6.2.3. Synthetic Rectangles

The respective segmentation landmark errors for all approaches comparing *landmark to landmark* and *landmark to contour* error are shown in Fig. 6.8. Boxplots of the *landmark to landmark* error for the five approaches can be found in Fig. 6.9. As expected all approaches performed well on this simple synthetic data set yielding mean landmark errors smaller than 2 pixel. This still remaining error result from rounding errors during image warps i.e. warp artifacts in the DE. Fig. 6.9(a) shows that the IMF approach yields small mean landmark errors around 0.5 pixel even when using only 1% of the region map. That allows the conclusion that the segmentation accuracy does not depend on the number of pixels used in the region map (e.g. the used percentage). The outliers appearing in all approaches on the one hand represent single miss segmentations during the LOOCV runs and on the other hand can again be warp artifacts.

The IMF approach finished the segmentation task first, before the two approaches using only regions inside the shape. The IMF approach was faster up to the following factors then all other approaches:  $3.6\times$  GaborIn,  $9.5\times$  GaborInOut,  $3.8\times$  MonoIn,  $10.4\times$  MonoInOut. This can be ascribed due to the fact that the IMF approach does not need the time consuming image feature classification step.

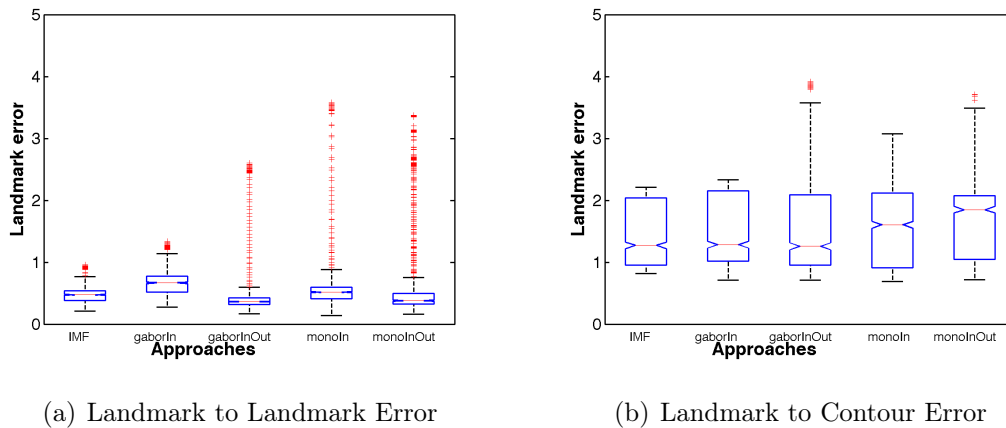
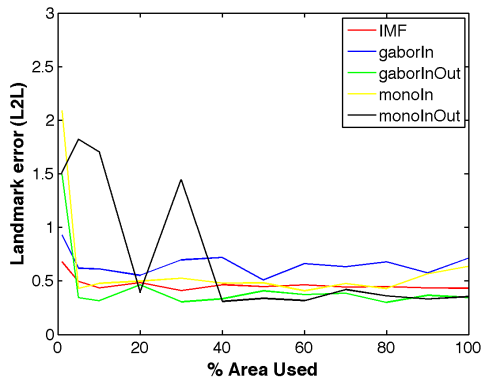
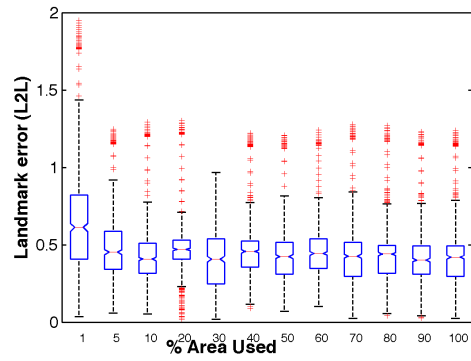


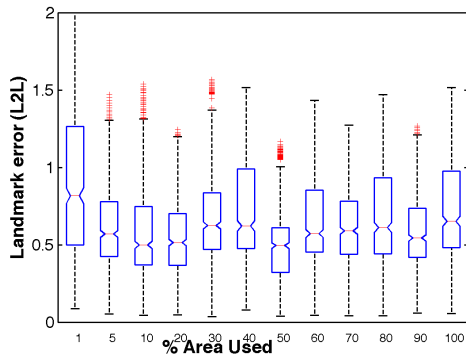
Figure 6.8.: Comparison boxplots of the *landmark to landmark* error (a) and the *landmark to contour* error (b) for each performed segmentation approach on the synthetic rectangles data using 100 percent of the region map.



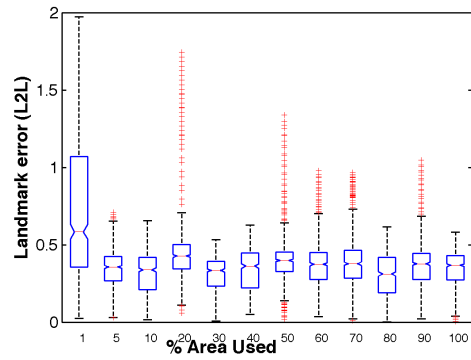
(a) Mean Errors for all Methods



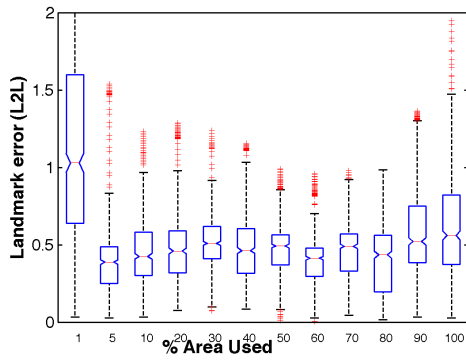
(b) IMF



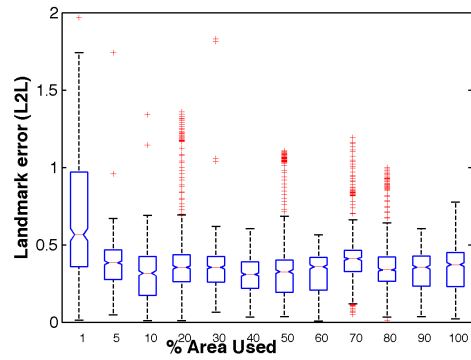
(c) GaborIn



(d) GaborInOut



(e) MonoIn



(f) MonoInOut

Figure 6.9.: Result plots for the synthetic rectangles data set showing the *landmark to landmark* error. The mean landmark error comparison of all applied segmentation approaches is shown in (a). Boxplots of the landmark error resulting when using a certain percentage of the respective region map are shown for the different approaches: IMF (b), GaborIn (c), GaborInOut (d), MonoIn (e), MonoInOut (f).

### 6.2.4. Metacarpal Bones

In Fig. 6.10 segmentation results with minimal and typical landmark error are presented and compared and in Fig. 6.11 the respective segmentation landmark errors for all approaches comparing *landmark to landmark* and *landmark to contour* error are shown.

All approaches using Monogenic features (IMF, MonoIn, MonoInOut) led to better segmentation results than those using Gabor Jets (GaborIn, GaborInOut). It can also be observed that the IMF approach yielded the lowest landmark error for all conducted segmentation runs. Furthermore it can be noticed that the segmentation accuracy apparently does not depend on the percentage of the region map used regardless of the approach (see Fig. 6.12). The reason therefore can be that due to the structural appearance of the inside and the surrounding of the object, the essential information needed for accurate segmentation can be incorporated into only a few pixel and even so equal segmentation results can be achieved. The obvious outliers in all approaches result from segmentation problems in the joint areas of the metacarpal bones. Especially when narrow joint areas between successive bones appear hardly any difference in structure and appearance can be extracted. It can be noted that despite this problems when using features based on the Monogenic Signal better segmentation results could be achieved than using Gabor Jets. This again is due to the better incorporation of edge and symmetry information achieved by applying the Monogenic Signal. A further problem is the previously mentioned rigid shape model, that constrains the shape generation during the DE. Therefore on objects with complex structure the segmentation accuracy is lower than on simple structured objects.

For the first four segmentation runs (using 1%, 5%, 10% and 20% of the region map) the IMF approach yielded the fastest results. Starting at 30% of the region map it was slower than the approaches using only inner regions (GaborIn, MonoIn). This can be reasoned by the fact that the image feature classification step is not influenced by the number of pixels per label of the region map, i.e. the classification runtime is independent of the region size but depends on the number of regions. Therefore when reaching a certain percentage of the region map the runtime loss due to the classification was compensated by longer runtimes due to a larger amount of pixels to process in the DE based segmentation. Before 30% of used pixel in the region map are reached, the IMF approach was faster then all other approaches up to the following factor:  $5.3\times$  GaborIn,  $44.0\times$  GaborInOut,  $3.4\times$  MonoIn,  $41.2\times$  MonoInOut. After reaching 30% of

pixel in the region map used the IMF approach was still faster than all approaches using inner and outer regions.

Overall can be noted that the IMF approach provided at least equal or even better segmentation accuracy on the metacarpal bones data set in comparison to all other analyzed approaches.

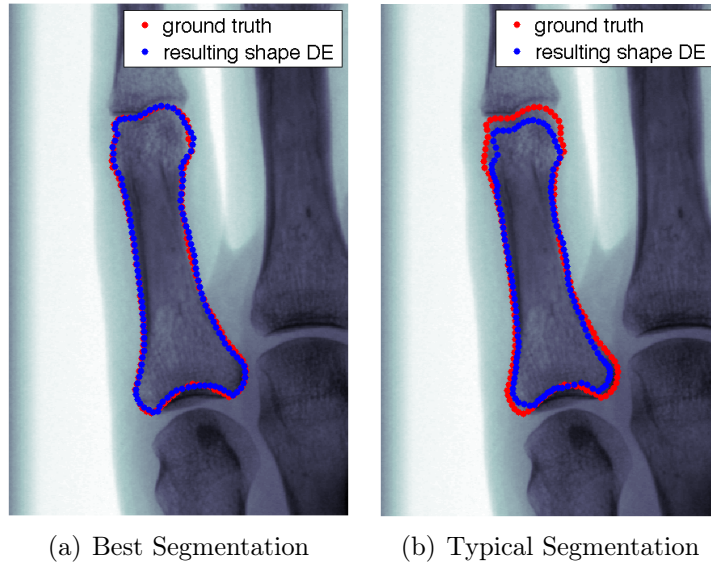


Figure 6.10.: Example segmentation results with minimal and typical mean landmark error for the metacarpal bones data set are shown.

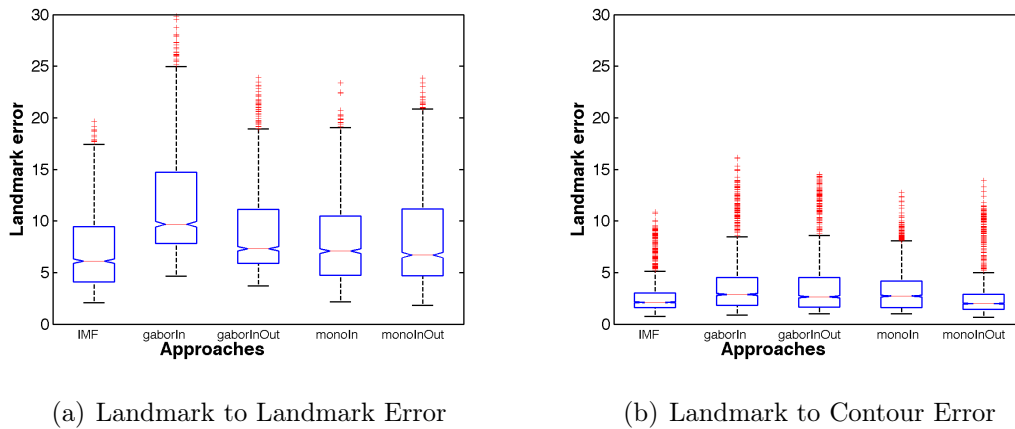
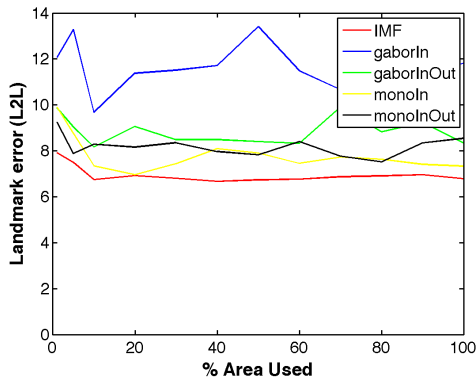
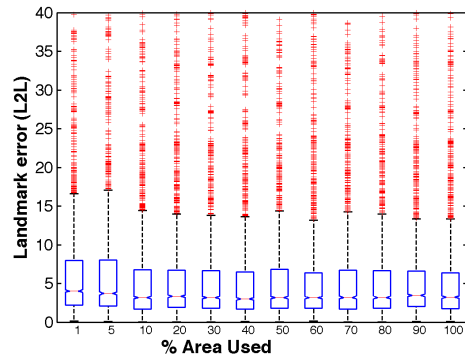


Figure 6.11.: In this figure comparison boxplots of the *landmark to landmark* error (a) and the *landmark to contour* error (b) for each performed segmentation approach on the metacarpal bones data set are shown.

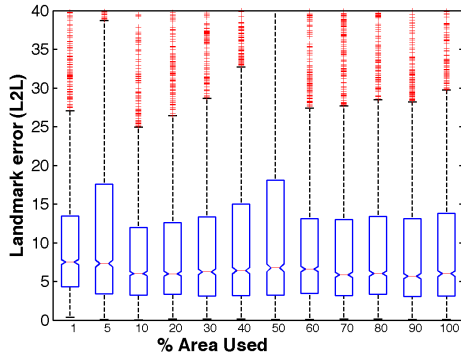




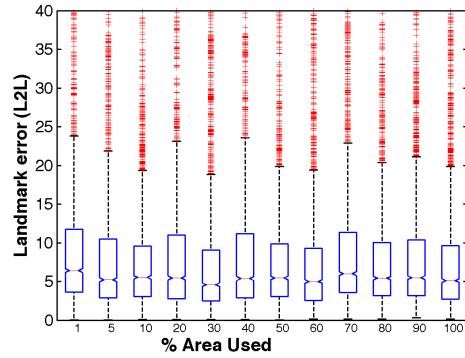
(a) Mean Errors for all Methods



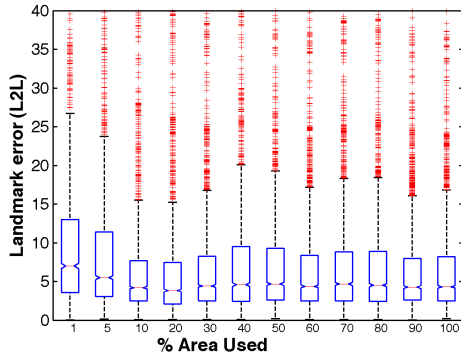
(b) IMF



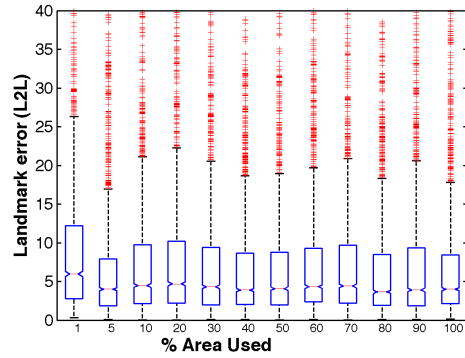
(c) GaborIn



(d) GaborInOut



(e) MonoIn



(f) MonoInOut

Figure 6.12.: Result plots for the metacarpal bones data set showing the *landmark to landmark* error. The mean landmark error comparison of all applied segmentation approaches is shown in (a). Boxplots of the landmark error resulting when using a certain percentage of the respective region map are shown for the different approaches: IMF (b), GaborIn (c), GaborInOut (d), MonoIn (e), MonoInOut (f).

### 6.2.5. Hearts

Segmentation results with minimal and typical landmark error are presented and compared in Fig. 6.13. The respective segmentation landmark errors for all approaches comparing *landmark to landmark* and *landmark to contour* error are shown in Fig. 6.14.

Taking a closer look at the result plots reveals that all approaches using Monogenic features (IMF, MonoIn, MonoInOut) outperformed those using Gabor Jets (GaborIn, GaborInOut) and even yielded less outliers (see Fig. 6.15). The outliers appearing in the approaches using Gabor Jets result from images where the heart and the surrounding area have similar gray values. This again reinforces the fact that Gabor Jets are prone to bad image quality as shadows caused during exposure, noisy images or low gray value differences, where Monogenic Signal based features deliver promising results. Furthermore it can be noticed that the segmentation accuracy apparently does not depend on the used percentage of the region map regardless of the approach except for GaborIn, that showed a landmark error increase with 1% and 5% of the region map. Again distinct appearance information is incorporated in only a few pixels of the region map rendering large region maps useless.

It can be observed that the runtime is increased proportionally to an increase of the percentage of the respective region map, regardless of the selected segmentation approach. As with the synthetic rectangles the IMF approach finished the segmentation task first, before the two approaches using only regions inside the shape. Generally can be said that approaches in combination with Monogenic features converged faster than those using Gabor Jets. The IMF approach was faster than all other approaches up to a following factor:  $7.5\times$  GaborIn,  $15.1\times$  GaborInOut,  $4.7\times$  MonoIn,  $11\times$  MonoInOut.

Overall can be noted that the IMF approach performed best in terms of computational speed and provided at least equal or even better segmentation accuracy on the hearts data set in comparison to all other analyzed approaches.

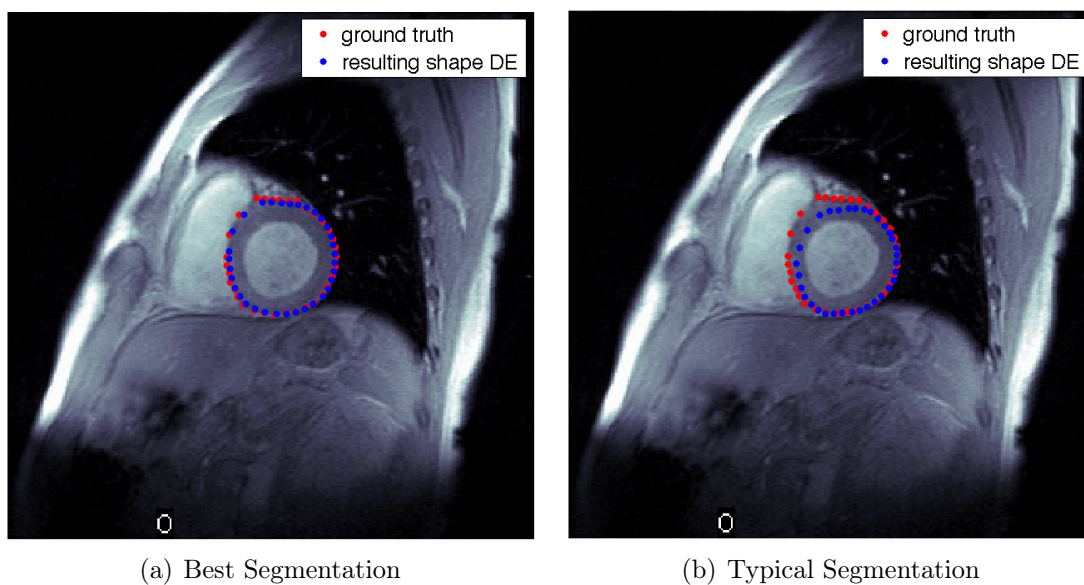


Figure 6.13.: Example segmentation results with minimal and typical mean landmark error for the hearts data set are shown.

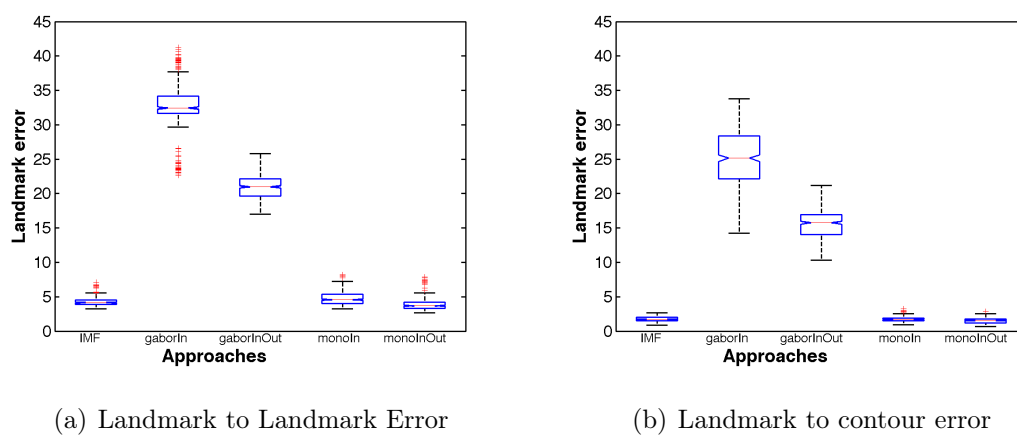
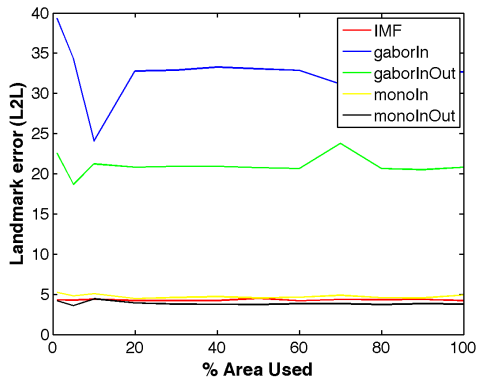
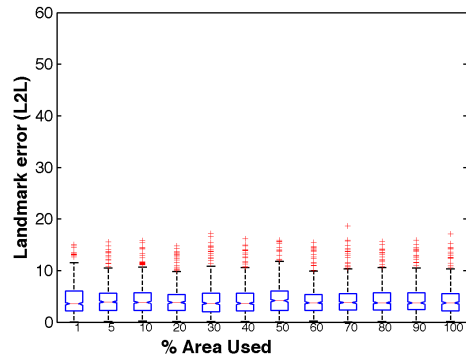


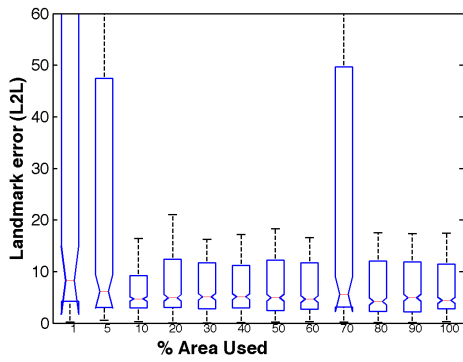
Figure 6.14.: In this figure comparison boxplots of the *landmark to landmark* error (a) and the *landmark to contour* error (b) for each performed segmentation approach on the hearts data set are shown.



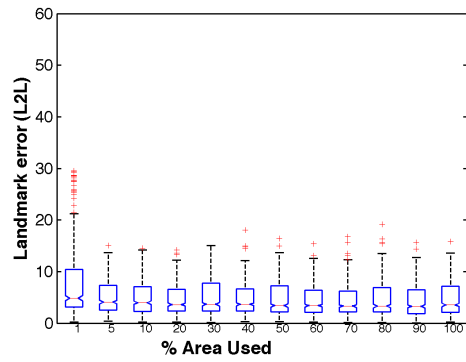
(a) Mean Errors for all Methods



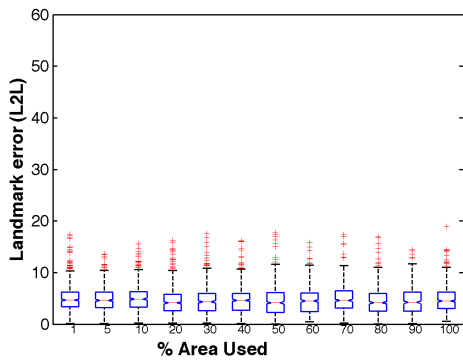
(b) IMF



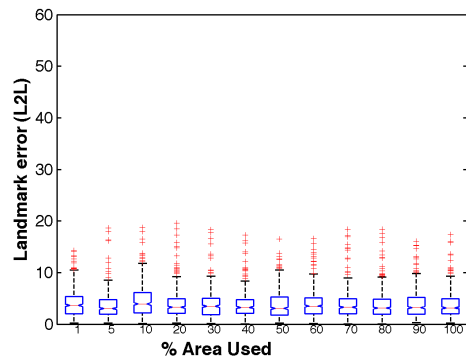
(c) GaborIn



(d) GaborInOut



(e) MonoIn



(f) MonoInOut

Figure 6.15.: Result plots for the hearts data set showing the *landmark to landmark* error. The mean landmark error comparison of all applied segmentation approaches is shown in (a). Boxplots of the landmark error resulting when using a certain percentage of the respective region map are shown for the different approaches: IMF (b), GaborIn (c), GaborInOut (d), MonoIn (e), MonoInOut (f).

### 6.2.6. Lungs Left

In Fig. 6.16 segmentation results minimal and typical landmark error are presented and compared. Taking a closer look at the result plots reveals that the approaches using Monogenic Signal based features yielded similar landmark errors, but again the IMF approach yielded approximately 1 to 3 pixels smaller landmark errors. The two approaches using Gabor Jets led to 1 to 2 pixel larger landmark errors then their equivalents using the Monogenic Signal. The lung represents a complex anatomical object and as with the metacarpal bones the applied rigid shape model can be blamed for inaccuracies in the segmentation results. The respective segmentation landmark errors for all approaches comparing *landmark to landmark* and *landmark to contour* error are shown in Fig. 6.17. Boxplots of the *landmark to landmark* error for the five approaches can be found in Fig. 6.18.

By observing the runtimes using different percentages of the region map, again the time increases with increasing percentages. Worth mentioning is the fact that on this data set the IMF approach was outperformed in terms of computational speed when reaching 50% of the region map by the approaches using only the region inside the shape. This can be reasoned by the dependency of the classification step on the numbers of regions as well as the dependency of the DE (the image warp) on the number of pixels similar to the results on the metacarpal bones. Before 50% of used pixel in the region map are reached, the IMF approach was faster then all other approaches up to a following factor:  $7.5\times$  GaborIn,  $12.5\times$  GaborInOut,  $3.7\times$  MonoIn,  $10.4\times$  MonoInOut

After reaching 50% of pixel in the region map used the IMF approach was still faster than all approaches using inner and outer regions. Compared to the approaches using only inner regions it was slower.

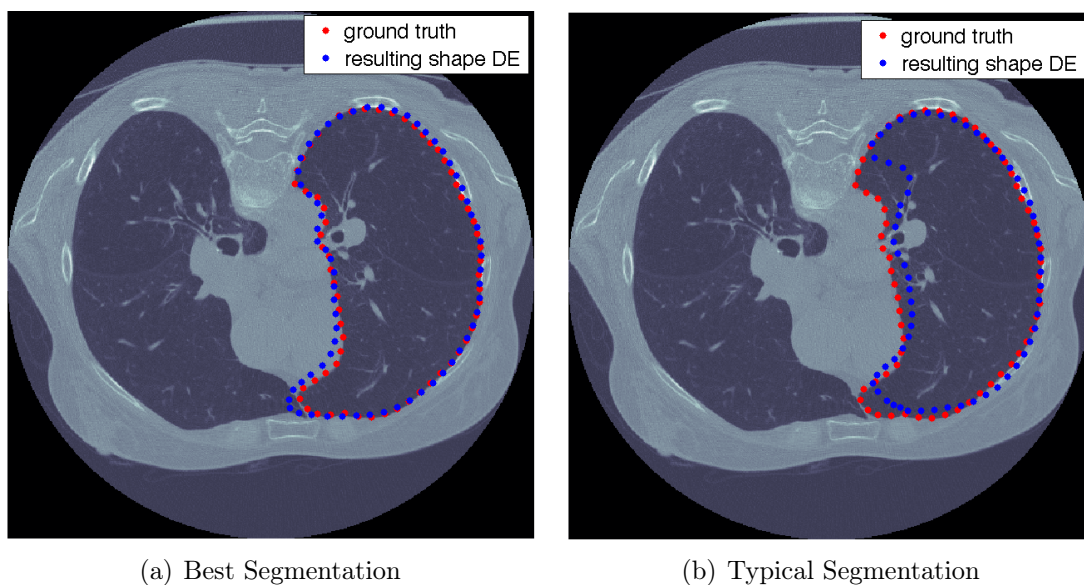


Figure 6.16.: Example segmentation results with minimal and typical mean landmark error for the lungs data set focusing on the left lung are shown.

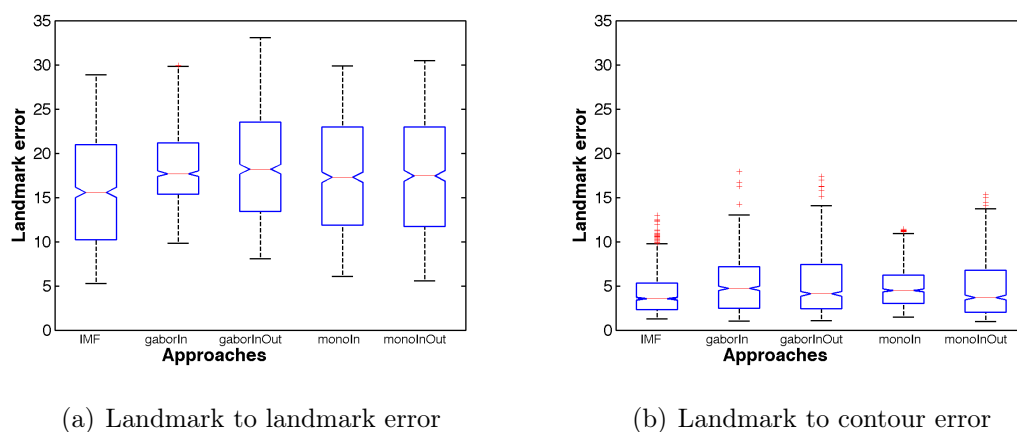
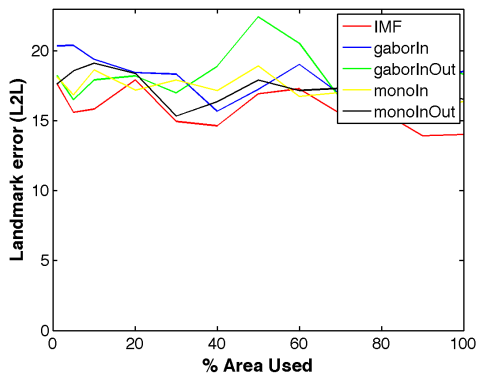
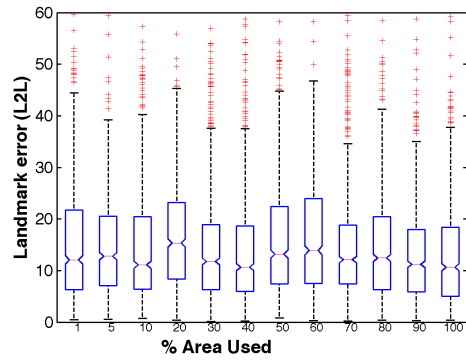


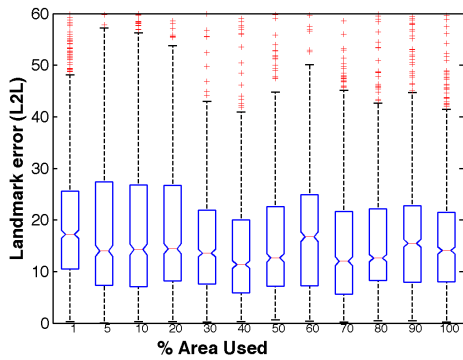
Figure 6.17.: In this figure comparison boxplots of the *landmark to landmark* error (a) and the *landmark to contour* error (b) for each performed segmentation approach on the lungs data set focusing on the left lung are shown.



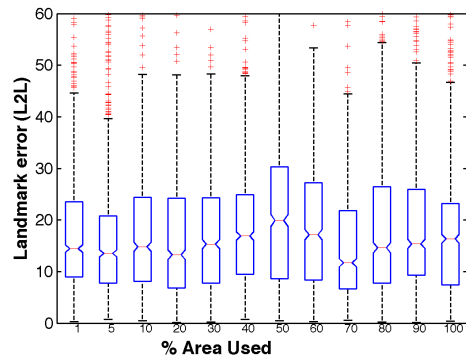
(a) Mean Errors for all Methods



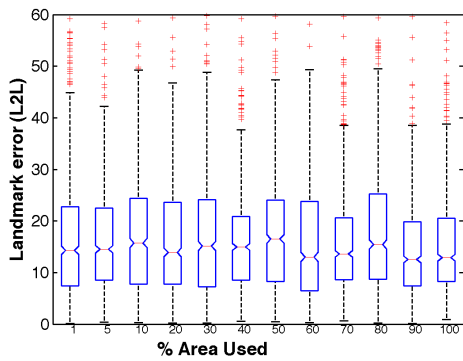
(b) IMF



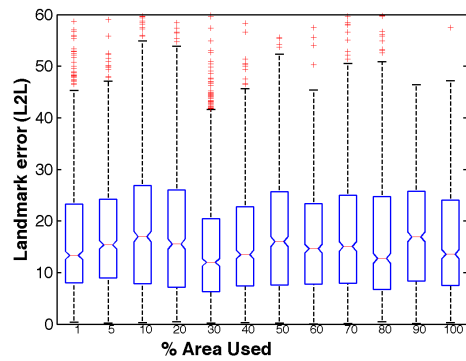
(c) GaborIn



(d) GaborInOut



(e) MonoIn



(f) MonoInOut

Figure 6.18.: Result plots for the lungs data set focusing on the left lung showing the *landmark to landmark* error. The mean landmark error comparison of all applied segmentation approaches is shown in (a). Boxplots of the landmark error resulting when using a certain percentage of the respective region map are shown for the different approaches: IMF (b), GaborIn (c), GaborInOut (d), MonoIn (e), MonoInOut (f).

### 6.2.7. Lungs Right

The results on this the right lung are similar to those on the left lung including differences in segmentation accuracy as well as computational performance. The respective segmentation landmark errors for all approaches comparing *landmark to landmark* and *landmark to contour* error are shown in Fig. 6.20. Boxplots of the *landmark to landmark* error for the five approaches can be found in Fig. 6.21. In Fig. 6.19 segmentation results minimal and typical landmark error are presented and compared.

The result plots reveals that the approaches using Monogenic Signal based features yielded similar landmark errors. The two approaches using Gabor Jets led to 1 to 5 pixel larger landmark errors then their equivalents using the Monogenic Signal.

By observing the runtimes using different percentages of the region map, again the time increases with increasing percentages. The IMF approach was this time outperformed in terms of computational speed when reaching 20% of the region map by the approaches using only the region inside the shape. Before 20% of used pixel in the region map are reached, the IMF approach was faster then all other approaches up to a following factor:  $8.0\times$  GaborIn,  $11.6\times$  GaborInOut,  $4.0\times$  MonoIn,  $11.2\times$  MonoInOut

After reaching 20% of used pixel in the region map the IMF approach was still faster than all approaches using inner and outer regions. Compared to the approaches using only inner regions it was slower.



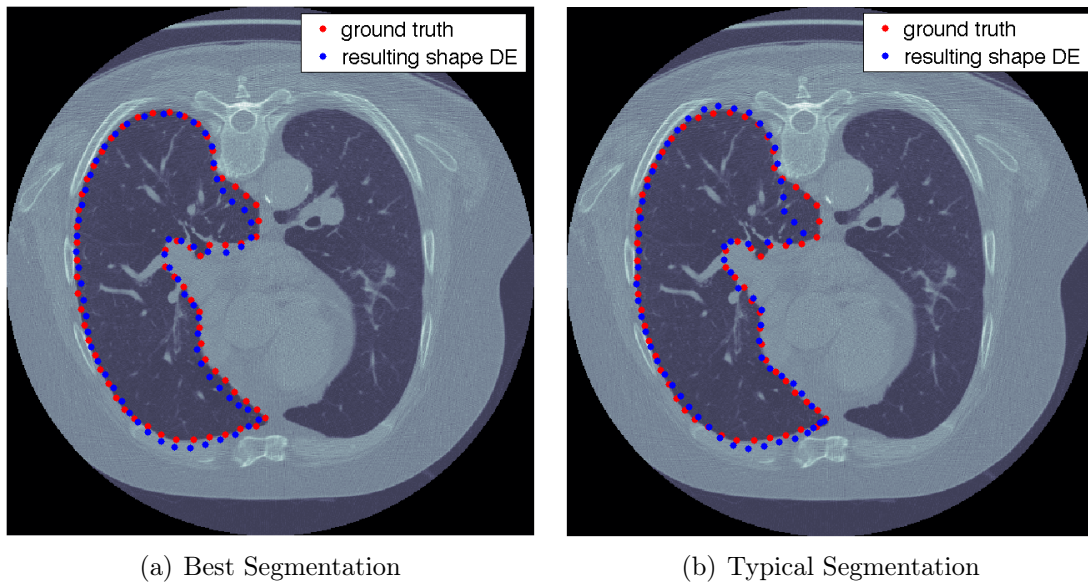


Figure 6.19.: Example segmentation results with minimal and typical mean landmark error for the lungs data set focusing on the right lung are shown.

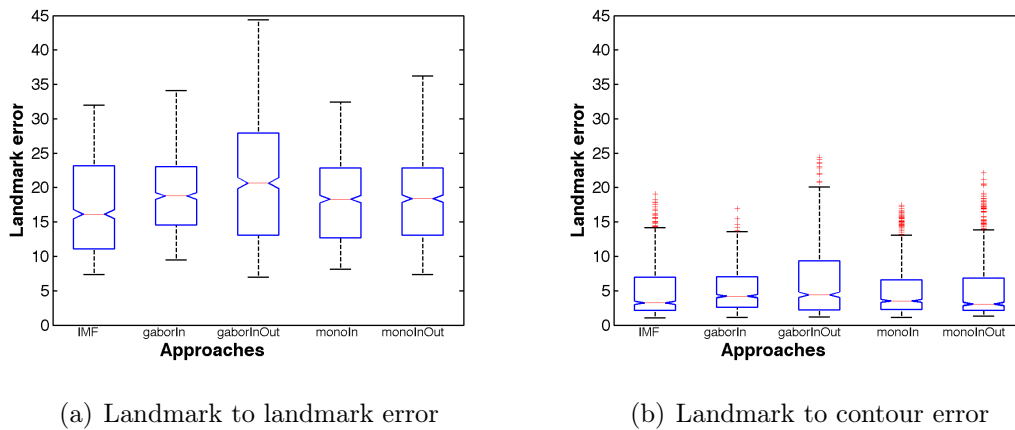
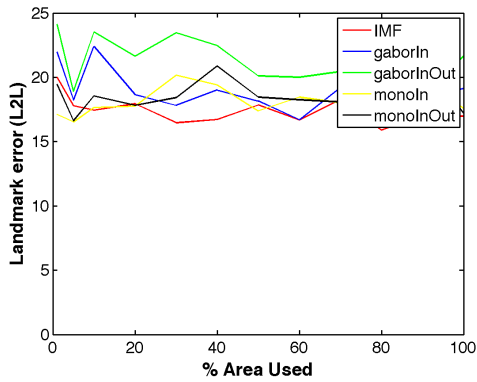
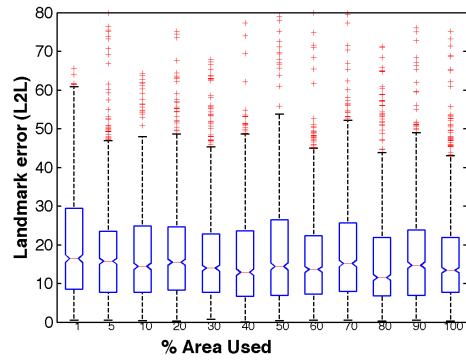


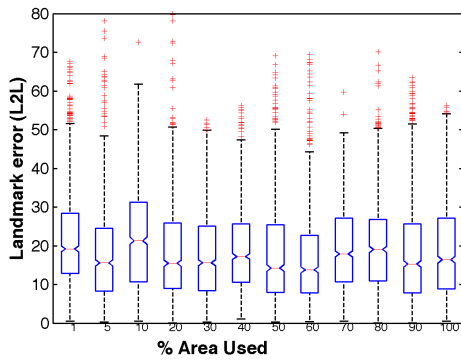
Figure 6.20.: In this figure comparison boxplots of the *landmark to landmark* error (a) and the *landmark to contour* error (b) for each performed segmentation approach on the lungs data set focusing on the right lung are shown.



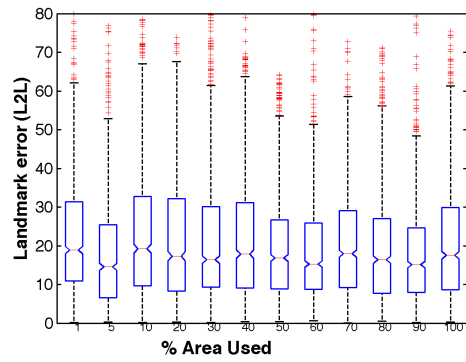
(a) Mean Errors for all Methods



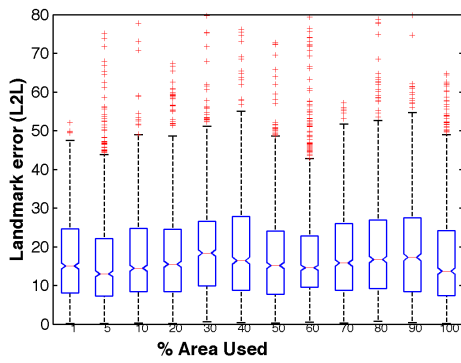
(b) IMF



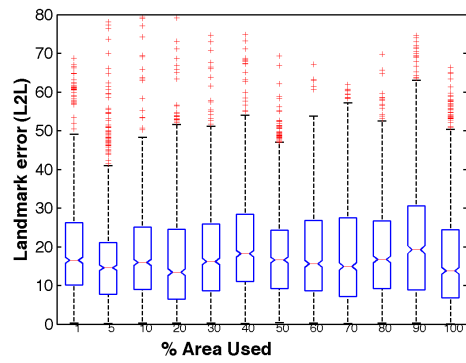
(c) GaborIn



(d) GaborInOut



(e) MonoIn



(f) MonoInOut

Figure 6.21.: Result plots for the lungs data set focusing on the right lung showing the *landmark to landmark* error. The mean landmark error comparison of all applied segmentation approaches is shown in (a). Boxplots of the landmark error resulting when using a certain percentage of the respective region map are shown for the different approaches: IMF (b), GaborIn (c), GaborInOut (d), MonoIn (e), MonoInOut (f).

## 6.3. Comparison with ASMs and AAMs

To be able to draw conclusions about the practicability of the proposed improved Shape Particle Filtering scheme, the results were compared with an Active Appearance Model (AAM) implementation proposed by [Donner06] and an Active Shape Model (ASM) implementation proposed by [Langs09]. [Donner06] enhanced AAMs using Canonical Correlation Analysis (CCA). For implementation details on these approaches please refer to the referenced papers.

The three algorithms differ in the following points:

- **Incorporated image information:** AAMs and Shape Particle Filters use shape and texture information. ASMs use only shape constraints and edge/image information near landmarks along so called profiles.
- **Capture range:** ASMs and AAMs use around 30% of the searched image [Cootes01]. Both approaches need at least a pre-initialized starting position overlapping two thirds of the object of interest. That means ASMs and AAMs need a manual localization of the object in the image prior to the actual segmentation. Shape Particle Filters offer automatic localization and segmentation of an object, without the need of any user interaction.

The mean landmark error resulting from the standard AAMs and their CCA based variant, the segmentation results based on ASMs ([Langs09]) and the results achieved in the scope of this thesis using Shape Particle Filters and 10% of the region map are shown in Tab. 6.6. The improved Shape Particle Filter approach (6.7 pixel) achieved similar results than the AAMs (6.4 pixel), while the other approaches yielded larger landmark errors. The ASM approach (3.6 pixel) outperformed AAMs as well as the proposed Shape Particle Filter. This can be explained due to the fact, that they do not try to fit possible solutions onto the whole image, but only at areas bounded by certain profiles.

An important detail that has to be noted, is the fact that the ASM as well as the AAM method are semiautomatic segmentation approaches, where the object of interest has to be located before the actual segmentation. The proposed improved Shape Particle Filter scheme on the other hand is able to locate and segment an object of interest automatically, without any additional pre location step needed.

By summing up the comparison results it can be concluded that the proposed segmentation scheme offers automatic location and segmentation whilst yielding equal landmark errors compared with AAMs but larger errors than ASMs (3.1 pixel mean landmark error difference).

	mean landmark to landmark error
SPF GaborIn	9.7
SPF GaborInOut	8.2
SPF MonoIn	7.3
SPF MonoInOut	8.3
SPF IMF	<b>6.7</b>
AAMs Standard	7.8
AAMs CCA	<b>6.4</b>
ASMs	<b>3.6</b>

Table 6.6.: Comparison of the mean *landmark to landmark* error in pixel of the proposed Shape Particle Filter segmentation approaches, an Active Appearance Model implementation [Donner06] and an Active Shape Model implementation [Langs09].

# Chapter 7.

## Conclusion and Outlook

In this chapter a conclusion about the presented results achieved in the scope of this thesis (Sec. 7.1) is drawn. In Sec. 7.2 an outlook containing possible future improvements is given.

### 7.1. Conclusion

By summing up the results presented in the scope of this thesis the following conclusion can be drawn. Segmentation results on five distinct data sets showed the applicability of the proposed algorithm on different modalities. Even on objects with complex shapes promising segmentation results were achieved.

The main contribution of this thesis is twofold. On the one hand the standard feature extraction method, namely Gabor Jets are replaced by Monogenic Signal based features. On the other hand two optimizations of the previous manually defined region map, namely automatic region maps and per-pixel region maps are proposed.

The Monogenic Signal yields features that are invariant to changes in contrast and brightness and directly incorporate phase and direction information to edges. Furthermore the fact that the number of features only depends on the selected filter scale is exploited and therefore Monogenic Signal based features are superior to those using the previous Gabor Jets. Approaches using these features lead to better segmentation results on all different data sets regardless of the use of automatic or per-pixel region maps.

Segmentation results relying on automatically generated region maps allow to directly incorporate structural and appearance information in the classification step. Compared

to manual region maps, they yield regions describing the underlying image features and providing an accurate representation of the spacial representation of the object of interest. When comparing manual and automatic region maps in terms of computational performance, the clustering in the training phase needed to create the automatic regions yields a slower image segmentation. The bottle neck in terms of computational performance when using manual or automatic region maps compared to per-pixel region maps is definitely the needed classification step. Although the used SVM yielded faster results than the two compared approaches ( $k$ -NN and  $k$ -NN with  $kd$ -tree) there is still place for improvements.

The approach using the introduced per-pixel maps does not need the costly classification step yielding a considerable increase in computational performance and therefore the fastest segmentation compared to all other approaches. Furthermore the determination of the optimal set of regions i.e. the clustering can be omitted. An equal or even better segmentation accuracy compared to all other approaches could be achieved. The current results on scientific data sets offer promising segmentation accuracy, but lack the necessary robustness and precision needed for practical application. Improvements concerning especially a more detailed shape model as well as a refinement of the Differential Evolution segmentation process are therefore needed.

## 7.2. Outlook

The following improvements for Shape Particle Filters are expected to improve performance and accuracy:

- Using a shape model with better generalization would relax the constraints given by the spatial representation of the objects to segment.
- Incorporating additional elasticity into the shape model during the Differential Evolution process to refine newly generated shapes and push them towards the shape to segment similar to ASMs. This would lead to a combination of the advantages of both methods.
- Enhancing the classification speed by using a random forest [Breiman01] based classification strategy for manual and automatic region map approaches.
- Enhancing the segmentation speed by systematically parallelizing the existing code base and swapping computational costly tasks to the graphic board.
- Modifying the current 2D Shape Particle Filter to be able to cope with three dimensional data.

# Appendix A.

## Additional Material and Results

### A.1. Shape Models: Eigenmodes and Eigenvalues of the Used Data Sets

The eigenvalues, that were calculated using PCA during the shape model generation (see Sec. 3.3) of all data sets except the synthetic rectangles are shown in Fig. A.1. The eigenvalues are sorted in descending order. 95 % of the data point variance is covered by the red bars. The calculated eigenmodes for the metacarpal bones and the hearts data set are shown in Fig. A.2 and for the two lungs data sets in Fig. A.3.



Appendix A. Additional Material and Results

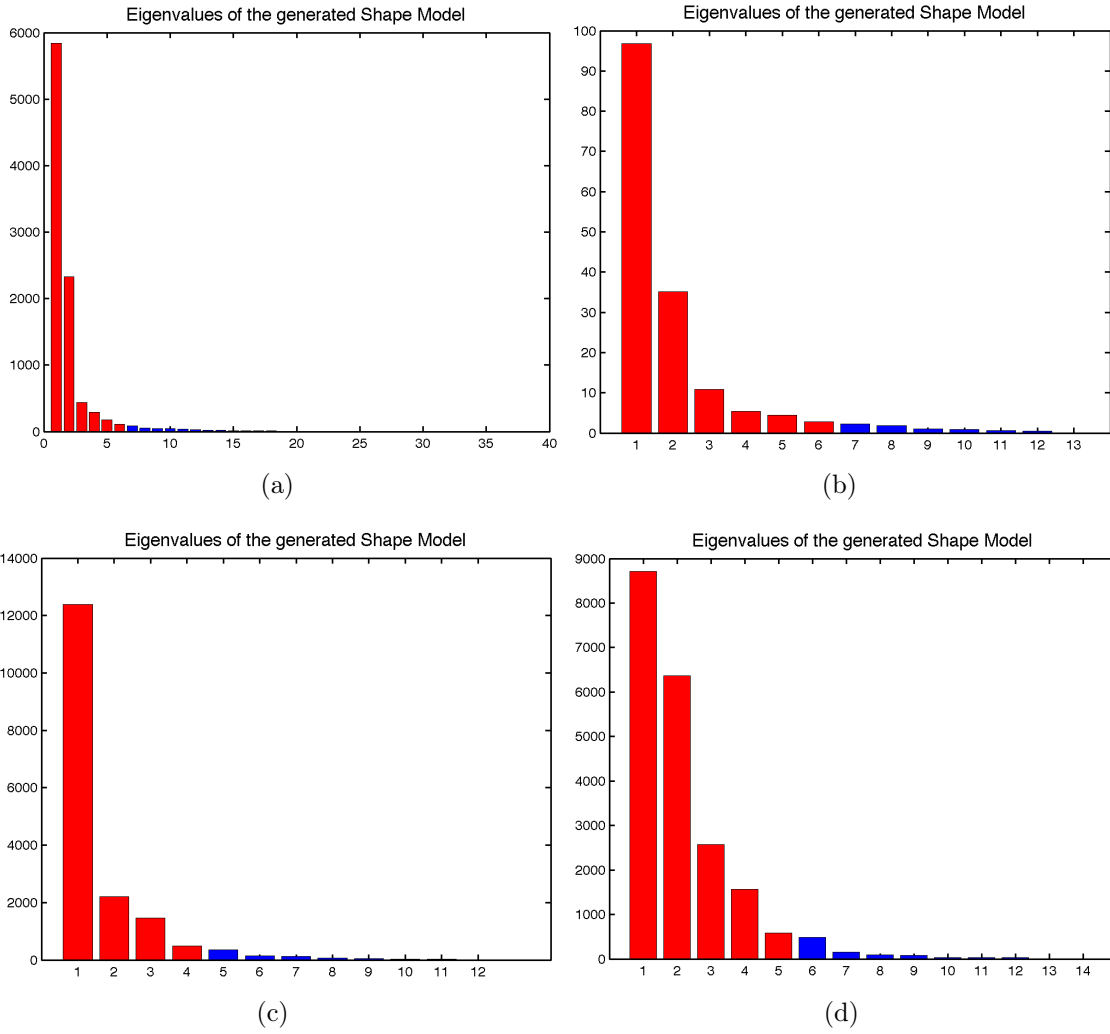


Figure A.1.: Bar diagrams of the calculated eigenvalues of the metacarpal bones (a), the hearts (b), the lungs focusing on the left (c) and on the right (d) lung. The red bars represent 95% of the data point variance.

## Appendix A. Additional Material and Results

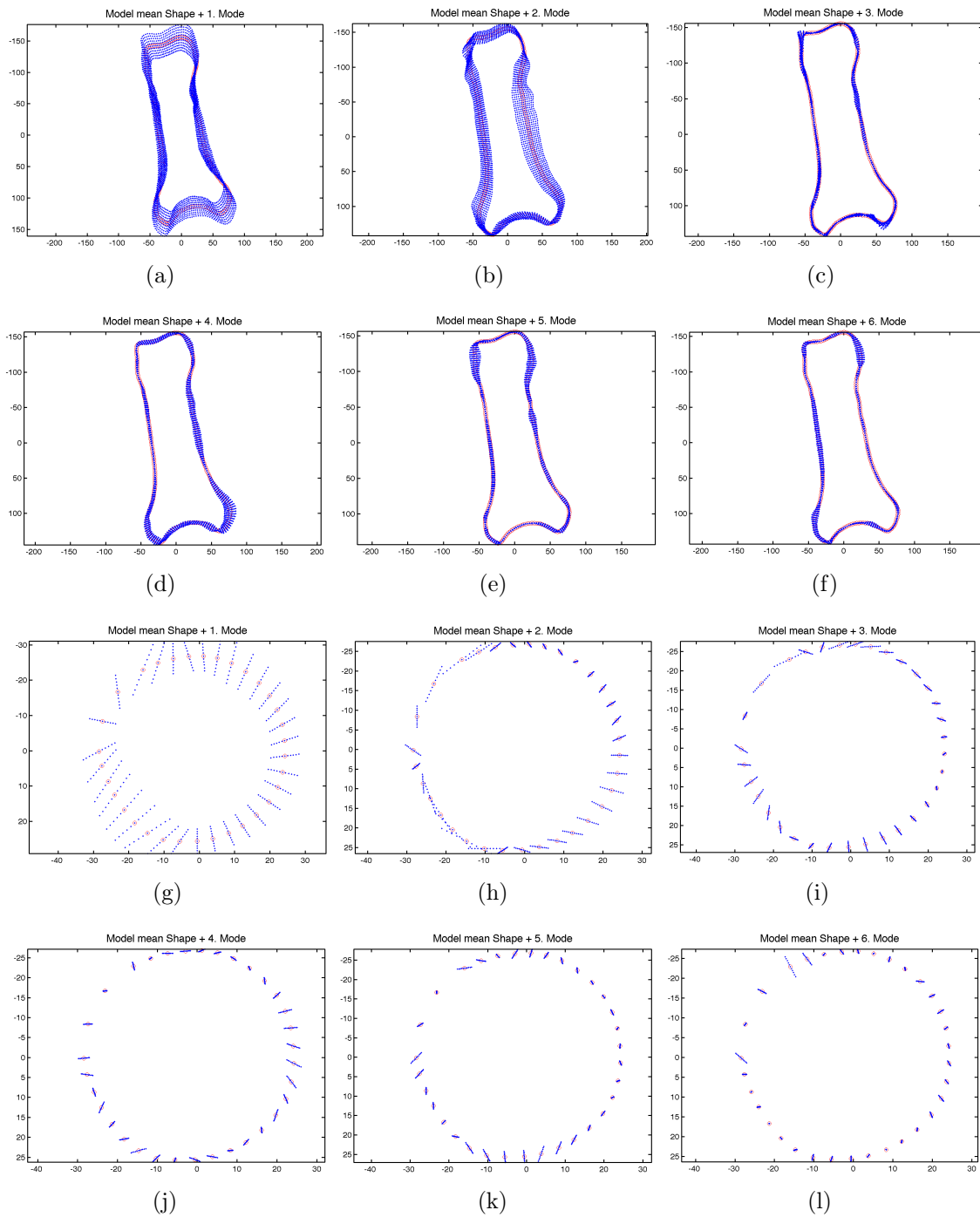


Figure A.2.: Visualization of the six calculated eigenmodes for the metacarpal bones (a) - (f) and the hearts (g) - (l) data sets. The mean shape is drawn in red, the modes are blue.

Appendix A. Additional Material and Results

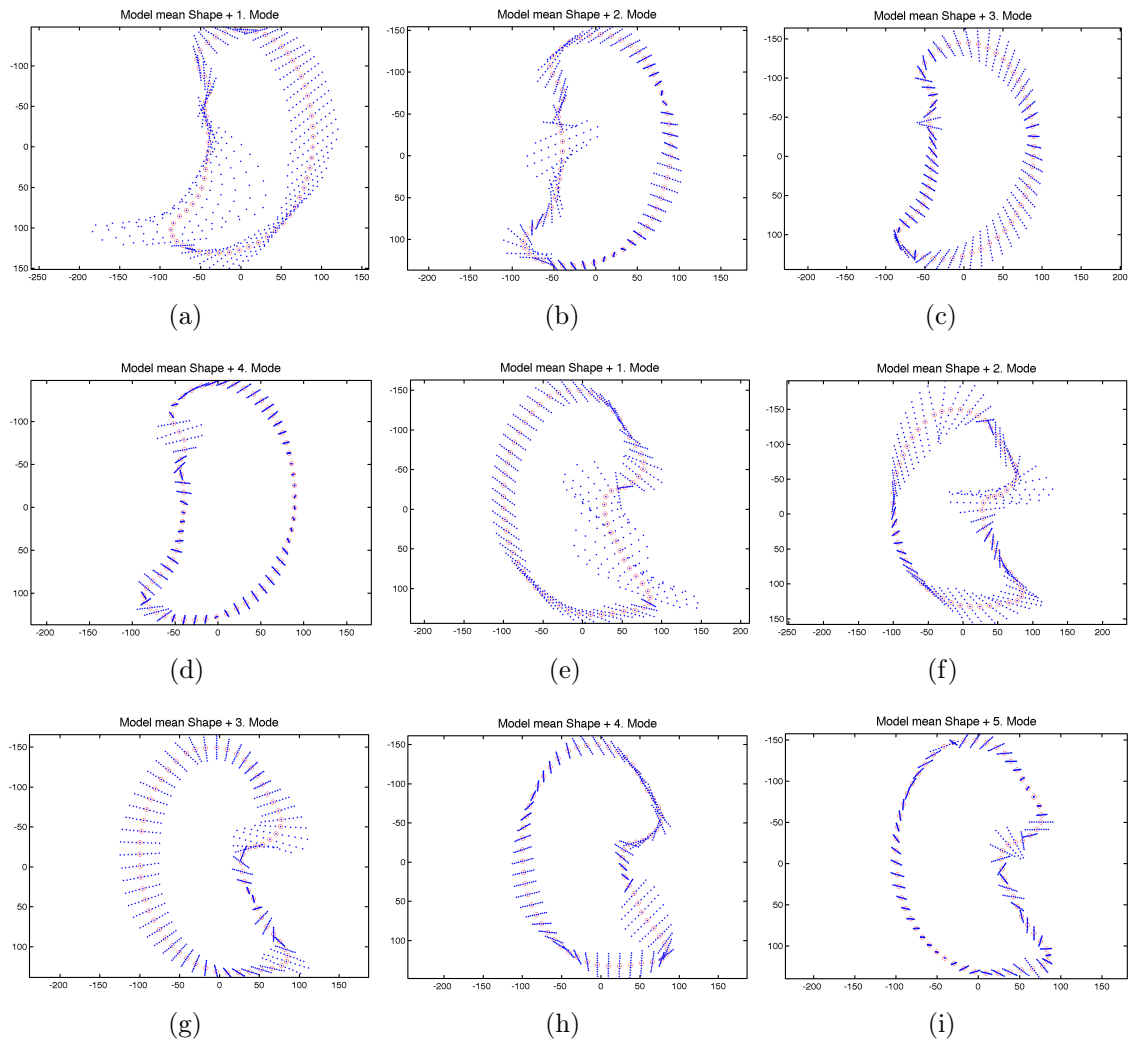


Figure A.3.: Visualization of the six calculated eigenmodes for the lungs data set focusing on the left (a) - (d) and the right lung (e) - (i). The mean shape is drawn in red, the modes are blue.

# Erklärung zur Verfassung der Arbeit

Lukas Fischer  
Liechtensteinstraße 143-145/13  
1090 Wien

“Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.”

Wien, 25. März 2010

# Bibliography

- [Andrieu03] C. Andrieu. Monte Carlo Methods for Absolute Beginners. In O. Bousquet, U. von Luxburg, G. Rätsch (Editors) *Advanced Lectures on Machine Learning*, vol. 3176 of *Lecture Notes in Computer Science*, pp. 113–145. Springer, 2003. ISBN 3540231226.
- [Arulampalam02] M. S. Arulampalam, S. Maskell, N. Gordon. A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. *IEEE Transactions on Signal Processing*, vol. 50(2):pp. 174–188, 2002.
- [Bentley75] J. Bentley. Multidimensional Binary Search Trees Used for Associative Searching. *Communications of the ACM*, vol. 18(9):pp. 509 – 517, 1975.
- [Bishop07] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1st ed. 2006. corr. 2nd printing ed., 2007. ISBN 0387310738.
- [Bookstein89] F. Bookstein. Principal Warps: Thin-Plate Splines and the Decomposition of Deformations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11(6):pp. 567 – 585, 1989.
- [Braak06] C. Braak. A Markov Chain Monte Carlo Version of the Genetic Algorithm Differential Evolution: Easy Bayesian Computing for Real Parameter Spaces. *Stat Comput*, vol. 16(3):pp. 239–249, 2006.
- [Breiman01] L. Breiman. Random Forests. *Machine Learning*, vol. 45(1):pp. 5–32, 2001. ISSN 08856125.
- [Burges98] C. J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, vol. 2(2):pp. 121–167, 1998.

- [Cootes92] T. Cootes, C. Taylor. Active Shape Models - 'Smart Snakes'. *In Proc. British Machine Vision Conference*, pp. 266–275, 1992.
- [Cootes01] T. F. Cootes, G. J. Edwards, C. J. Taylor. Active Appearance Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23(6):pp. 681–685, 2001. ISSN 0162-8828.
- [Cover67] T. Cover, P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, vol. 13(1):pp. 21–27, 1967.
- [deBruijne04a] M. de Bruijne, M. Nielsen. Image Segmentation by Shape Particle Filtering. *Proc. ICPR*, vol. 3:pp. 722–725, 2004.
- [deBruijne04b] M. de Bruijne, M. Nielsen. Shape Particle Filtering for Image Segmentation. *Proc. MICCAI 2004*, vol. 3216:pp. 168–175, 2004.
- [deBruijne05] M. de Bruijne, M. Nielsen. Multi-Object Segmentation Using Shape Particles. *Lecture Notes in Computer Science: Information Processing in Medical Imaging*, vol. 3565:pp. 762–773, 2005.
- [deBruijne06] M. de Bruijne. Shape Particle Guided Tissue Classification. *Mathematical Methods in Biomedical Image Analysis (MMBIA)*, p. 8, 2006.
- [Donner06] R. Donner, M. Reiter, G. Langs. Fast Active Appearance Model Search Using Canonical Correlation Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1690–1694, 2006.
- [Doucet01] A. Doucet, N. De Freitas, N. Gordon (Editors) *Sequential Monte Carlo Methods in Practice (Statistics for Engineering and Information Science)*. Springer, 1 ed., 2001. ISBN 0387951466.
- [Fearnhead08] P. Fearnhead. *Sequential Monte Carlo Methods in Filter Theory*. Ph.D. thesis, University of Oxford, 2008.
- [Felsberg01] M. Felsberg, G. Sommer. The Monogenic Signal. *IEEE Transactions on Signal Processing*, vol. 49(12):pp. 3136–3144, 2001.
- [Felsberg02] M. Felsberg. *Low-Level Image Processing with the Structure Multivector*. Ph.D. thesis, Institute of Computer Science and Applied Mathematics. Christian-Albrechts-University of Kiel, 2002.

- [Fischer09] L. Fischer, R. Donner, F. Kainberger, G. Langs. Automatic Region Template Generation for Shape Particle Filtering based Image Segmentation. *Proc. of the MICCAI 2009 Workshop on Probabilistic Models for Medical Image Analysis (PMMIA 2009)*, pp. 289–300, 2009.
- [Fix52] E. Fix, J. H. Jr. Discriminatory Analysis-Nonparametric Discrimination: Small Sample Performance. Tech. Rep. 11, USAF School of Aviation Medicine, Randolph Field, Texas, 1952.
- [Fix89] E. Fix, J. H. Jr. Discriminatory Analysis: Nonparametric discrimination: Consistency properties. Tech. Rep. 4, USAF School of Aviation Medicine, Randolph Field, Texas, 1989.
- [Florin05] C. Florin, N. Paragios, J. Williams. Particle Filters, a Quasi-Monte Carlo Solution for Segmentation of Coronaries. *Proc. MICCAI 2005*, vol. 3749:pp. 246–253, 2005.
- [Geman84] S. Geman, D. Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):pp. 721–741, 1984.
- [Giebel04] J. Giebel, D. Gavrilu, C. Schnorr. A Bayesian Framework for Multi-cue 3D Object Tracking. *Computer Vision - ECCV 2004*, pp. 241–252, 2004.
- [Gordon93] N. Gordon, D. Salmond, A. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEEE Proceedings for Radar and Signal Processing*, vol. 140(2):pp. 107–113, 1993.
- [Grunwald05] P. Grunwald, I. J. Myung, M. A. Pitt. Advances in minimum description length: theory and applications. *MIT Press*, p. 452, 2005.
- [Halton70] J. Halton. A retrospective and prospective survey of the Monte Carlo method. *Siam review*, 1970.
- [Hammersley64] J. M. Hammersley, D. C. Handscomb. *Monte Carlo methods*. Taylor Francis, 1964.
- [Kokkinos08] I. Kokkinos, A. Yuille. Scale Invariance without Scale Selection. *IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008*, pp. 1–8, 2008.

- [Langs09] G. Langs, P. Peloschek, H. Bischof. Automatic Quantification of Joint Space Narrowing and Erosions in Rheumatoid Arthritis. *IEEE Transactions on Medical Imaging*, vol. 28(1):pp. 151–164, 2009.
- [Lee03] J. Lee, J. Wang, C. Zhang. Color Image Segmentation: Kernel Do the Feature Space. In *Machine Learning: ECML 2003*, vol. 2837 of *Lecture Notes in Computer Science*, pp. 253–264. Springer Berlin / Heidelberg, 2003.
- [Liebelt07] J. Liebelt, K. Schertler, E. Germany. Precise Registration of 3D Models to Images by Swarming Particles. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, pp. 1–8, 2007.
- [MacKay03] D. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.
- [MacQueen67] J. MacQueen. Some methods for classification and analysis of multivariate observations. *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1:pp. 281–297, 1967.
- [Metropolis53] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. Teller. Equation of state calculations by fast computing machines. *The journal of Chemical Physics*, vol. 21(6), 1953.
- [Movellan96] J. Movellan. Tutorial on Gabor Filters. *Tutorial paper <http://mplab.ucsd.edu/tutorials/pdfs/gabor.pdf>*, 1996.
- [Pan06] X. Pan, M. Brady, R. Highnam, J. Declerck. The use of multi-scale monogenic signal on structure orientation identification and segmentation. *Lecture Notes in Computer Science*, vol. 4046:p. 601, 2006.
- [Pearson01] K. Pearson. On Lines and Planes of Closest Fit to Systems of Points in Space. *Philosophical Magazine*, vol. 2:pp. 559–572, 1901.
- [Rajpoot09] K. Rajpoot, V. Grau, J. Noble. Local-phase based 3d boundary detection using monogenic signal and its application to real-time 3-D echocardiography images. *Proc. of the Sixth IEEE international conference on Symposium on Biomedical Imaging: From Nano to Macro*, pp. 783–786, 2009.



- [Rissanen78] J. Rissanen. Modeling by shortest data description. *Automatica*, vol. 14:pp. 465–471, 1978.
- [Schölkopf00] B. Schölkopf, A. Smola, R. Williamson, P. Bartlett. New Support Vector Algorithms. *Neural Computation*, vol. 12(5):pp. 1207–1245, 2000.
- [Seise09] M. Seise, S. J. Mckenna, I. W. Ricketts, C. A. Wigderowitz. Parts-based segmentation with overlapping part models using Markov chain Monte Carlo. *Image and Vision Computing*, vol. 27(5):pp. 504–513, 2009.
- [Smith02] L. Smith. A tutorial on Principal Components Analysis. Tech. rep., Cornell University, 2002.
- [Sørensen08] L. Sørensen, J. Østergaard, P. Johansen, M. de Bruijne. Multi-object Tracking of Human Spermatozoa. *Proc. of SPIE*, vol. 6914(3):pp. 1–12, 2008.
- [Stegmann01] M. Stegmann, R. Fisker, B. Ersboll. Extending and Applying Active Appearance Models for Automated, High Precision Segmentation in Different Image Modalities. *Proc. 12th Scandinavian Conference on Image Analysis - SCIA 2001*, pp. 90–97, 2001.
- [Storn97] R. Storn, K. Price. Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization*, vol. 11(4), 1997.
- [Takaya07] K. Takaya. Feature Point Correspondence of Stereo Images by Monogenic Phase. *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, pp. 272–275, 2007.
- [Taylor92] C. J. Taylor, D. H. Cooper, J. Graham. Training models of shape from sets of examples. In *In Proc. British Machine Vision Conference*, pp. 9–18. 1992.
- [Thodberg03] H. H. Thodberg. Minimum Description Length Shape and Appearance Models. In *Information Processing in Medical Imaging*, vol. 2732 of *Lecture Notes in Computer Science*, pp. 51–62. 2003.
- [Vapnik95] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, 1995.

- [Wietzke08] L. Wietzke, G. Sommer. The Conformal Monogenic Signal. In *Pattern Recognition*, vol. 5096 of *Lecture Notes in Computer Science*, pp. 527–536. Springer Berlin / Heidelberg, 2008.
- [Wietzke09] L. Wietzke, G. Sommer, O. Fleischmann. The Conformal Monogenic Signal of Image Sequences. In *Statistical and Geometrical Approaches to Visual Motion Analysis: International Dagstuhl Seminar*, Lecture Notes in Computer Science, pp. 305–322. Springer Berlin / Heidelberg, 2009.
- [Wörz09] S. Wörz, W. Godinez, K. Rohr. Probabilistic Tracking and Model-Based Segmentation of 3D Tubular Structures. *Bildverarbeitung Für Die Medizin 2009: Algorithmen-Systeme-Anwendungen*, p. 41, 2009.
- [Yoshimura00] H. Yoshimura, M. Etoh, K. Kondo, N. Yokoya. Gray-scale character recognition by Gabor jets projection. *Proc. ICPR*, vol. 2:pp. 335–338, 2000.